Graduate School ETD Form 9
(Revised 12/07)

PURDUE UNIVERSITY
GRADUATE SCHOOL
Thesis/Dissertation Acceptance

This is to certify that the thesis/dissertation prepared

By Ajay Kumar Bandi

Entitled
AN INTEGRATED SENSOR SYSTEM FOR EARLY FALL DETECTION

For the degree of Master of Science in Electrical and Computer Engineering

Is approved by the final examining committee:

Maher Rizkalla
Chair

Paul Salama

Dongsoo Stephen Kim

To the best of my knowledge and as understood by the student in the *Research Integrity and Copyright Disclaimer (Graduate School Form 20)*, this thesis/dissertation adheres to the provisions of Purdue University's "Policy on Integrity in Research" and the use of copyrighted material.

Approved by Major Professor(s): Maher Rizkalla

Approved by: Brian King 4/5/2013
Head of the Graduate Program Date

AN INTEGRATED SENSOR SYSTEM FOR EARLY FALL DETECTION

A Thesis

Submitted to the Faculty

of

Purdue University

by

Ajay Kumar Bandi

In Partial Fulfillment of the

Requirements for the Degree

of

Master of Science in Electrical and Computer Engineering

May 2013

Purdue University

Indianapolis, Indiana

To my family.

ACKNOWLEDGMENTS

I express my heart felt thanks and sincere gratitude to my thesis advisor, Dr. Maher Rizkalla for his assistance and guidance during the entire course of my masters. Special thanks to Dr. Alfredo Lopez for giving me this wonderful opportunity to work with him. I would like to thank all the engineers at Technicolor who shared their invaluable experience with me during my internship which was very much helpful for my thesis.

I would like to thank my friends Diana, Swetcha, Ravi, Abhiram, and Shantan for their encouragement and help, through the duration of my thesis. I would also like to thank Valerie Lim Diemer, Sherrie Tucker, and Jeff Sears for their help.

TABLE OF CONTENTS

Page

LIST OF FIGURES

ABSTRACT

Bandi, Ajay Kumar. M. S. E. C. E. , Purdue University, May 2013. An Integrated Sensor System for Early Fall Detection. Major Professor: Maher E. Rizkalla.

Physical activity monitoring using wearable sensors give valuable information about patient's neuro activities. Fall among ages of 60 and older in US is a leading cause for injury-related health issues and present serious concern in the public health care sector. If the emergency treatments are not on time, these injuries may result in disability, paralysis, or even death. In this work, we present an approach that early detect fall occurrences. Low power capacitive accelerometers incorporated with microcontroller processing units were utilized to early detect accurate information about fall events. Decision tree algorithms were implemented to set thresholds for data acquired from accelerometers. Data is then verified against their thresholds and the data acquisition decision unit makes the decision to save patients from fall occurrences. Daily activities are logged on an onboard memory chip with Bluetooth option to transfer the data wirelessly to mobile devices.

In this work, a system prototype based on neurosignal activities was built and tested against seven different daily human activities for the sake of differentiating between fall and non-fall detection. The developed system features low power, high speed, and high reliability. Eventually, this study will lead to wearable fall detection system that serves important need within the health care sector.

In this work Inter-Integrated Circuit (I2C) protocol is used to communicate between the accelerometers and the embedded control system. The data transfer from the Microcontroller unit to the mobile device or laptop is done using Bluetooth technology.

1. INTRODUCTION

The fall is a very risky factor in elderly people's daily living, especially the independently living elders; it often causes serious physiological injuries, such as bleeding, fracture, and central nervous system damages. If the emergency treatments are not on time, these injuries may result in disability, paralysis or even death. On the other hand, the fall may produce many psychological problems such as fear of movement, and worry about living independently [1]. It is estimated that over one third of adults of ages 60 years and older fall each year, making it a leading cause of nonfatal injury for that age group.

In 2002, about 22% of community-dwelling seniors reported falling events with medicare costs per fall averaged between $9,113 and $13,507 [2]. In 2000, falls among older adults cost the U.S. health care system over $19 billion and this reached $30 billion by 2010. With the population aging, both the number of falls and the costs to treat fall injuries are likely to increase. By 2020, the annual direct and indirect cost of fall injuries is expected to reach $54.9 billion [3].

One in three adults of age 65 and older is subjected to a fall each year [4] [5]. Of those, 20% to 30% suffer moderate to severe injuries that make it hard to live independently, and increase their risk of early death [6]. Older adults are hospitalized for fall-related injuries five times more than they are for injuries from other causes. In 2009, about 20,400 older adults died from unintentional fall injuries [5]. In the same year emergency departments treated 2.4 million nonfatal injuries among older adults; more than 662,000 of those patients were hospitalized [7].

1.1 Advantages of Fall Detection

Most of the present clinical assessment tools include either self-report or observer-rated measures. Although self-report measures are simple to acquire, they could be inaccurate due to many reasons such as poor patient memory and misjudgments of actual capability. Observer-related surveys by personal caretakers are often time consuming and rarely capture changes in functional status that may fluctuate throughout the day. Remote monitoring of physical activity using body-worn sensors provides an alternative to assessment of functional independence by subjective and paper-based questionnaires. The objectivity and comprehensiveness of a patients physical performance record could be improved by a system that automatically identifies the activities carried out by the individual throughout the day, particularly in remote locations such as the patients home or community.

1.2 Literature Review for Fall Detection

There have been research efforts to detect fall events. In an article written on Barometric pressure and triaxial accelerometry-based falls event detection [8], a waist mounted device was designed to feature a barometric sensor for the reduction of false positives in fall events using decision tree for pattern recognition. The device comprises of a custom-made data acquisition device, which comprises of a triaxial accelerometer (MMA7260, Freescale Sample Rate 40Hz), a microcontroller (MSP430F149, Texas Instruments) with a 12-bit analog-to-digital convertor (ADC), an atmospheric air pressure sensor (SCP1000, VTI Technologies), a Bluetooth module (WML-C30AH, Mitsumi), and a Li-Pol rechargeable battery. The data processing begins with incoming signals passing through median filter, then a low pass filter from where the gravitational acceleration component is obtained, and subtraction from the median filtered signal gives estimation of body acceleration. SVM (Signal Vector Magnitude from GA component), SMA (Signal Magnitude Area from BA component, tilt angle, and differential pressure parameter are extracted from the processed signal.

In an article about-automatic fall detection using wearable biomedical signal measurement terminal [9] discuss about a waist mounted device that detects fall events in real time with an alarm feature. The device uses a Kionix KXM52-1050 (Sample Frequency of 20Hz) tri-axial accelerometer and a Bellwave BSM856 CDMA (Code Division Multiple Access) standalone modem for detection and signal management. The fall detection has four thresholds (Fall Upper Threshold, Fall Lower Threshold, Laying Upper Threshold, Laying Lower Threshold) and uses only Y-axis to determine them.

With the advent of smartphones with accelerometers, programmability can be used to implement the fall detection algorithms directly in smartphones but as the phone may not be carried by the person at all times and the chances of the phone being dropped accidentally makes this option unfeasible [10].

1.3 Wireless Applications

A sensor is a device, which can convert physical information into signals, which can be interpreted by a user using an electronic component. Usually the signals received from these sensors are in analog form and can be converted and formatted into digital by using computers. With the advent of technology we can now use sensors, which are smart and efficient enough such that they come with all the processing and conversion units on the sensor body itself. These smart sensors are energy efficient and they also have embedded functions to communicate, transfer data and can also take inputs from the computers to accomplish the applications.

Smart sensors can be used to design integrated data acquisition systems [11], where they are used to obtain data continuously, process them and implement them in their respective applications to accomplish the tasks assigned. High-resolution data is expected to have from these sensors so that the uncompromised accuracies can be obtained. Sending these high-resolution data to the remote computers in real-time

gives the ability to monitor and store the data efficiently. It also reduces the size of the processing unit, which is supposed to be with the person at all times.

1.4 Embedded Sensor System Applications

Sensor technology has been used in measuring different physical quantities such as position, temperature, humidity, orientation, pressure, torque, radiation, acceleration and many more. With this wide range of capabilities, sensors find their applications in many areas in our day-to-day life. Applications include Medical, automotive, industrial, HVAC (heating, ventilation, and air conditioning), civilian etc. In this thesis we are primarily concerned about medical [12] and civilian usage of sensors to monitor and protect elders from fatal fall occurrences in real-time.

1.5 Sensor Fabrication Techniques

Semiconductors play a major role in sensor manufacturing using advanced techniques like MEMS (Micro-Electro Mechanical System), lab-on-chip, system-on-chip and ASIC (Application-Specific Integrated Circuit). These sensors are capable of doing data acquisition and signal processing at the same time consuming the lowest possible power. Figure1.1 explains the basic digital processing system inside a typical sensor. Initially, the physical data is obtained from the sensing area and the receiver section turns it into the digital signal-processing unit. Here the analog signals are converted to digital signals using A/D converters. The output of this block is then given to the transmitter section, where the data can be transmitted to other circuits like microprocessors or computers using various communication protocols, some of them include I2C, SPI and UART.

In this project we used accelerometers as the sensor units, which can record the patients physical activity. There are different types of accelerometers and what differentiates them is the type of sensing element and the principle of operation involved. The following is the list of typical accelerometers in use:

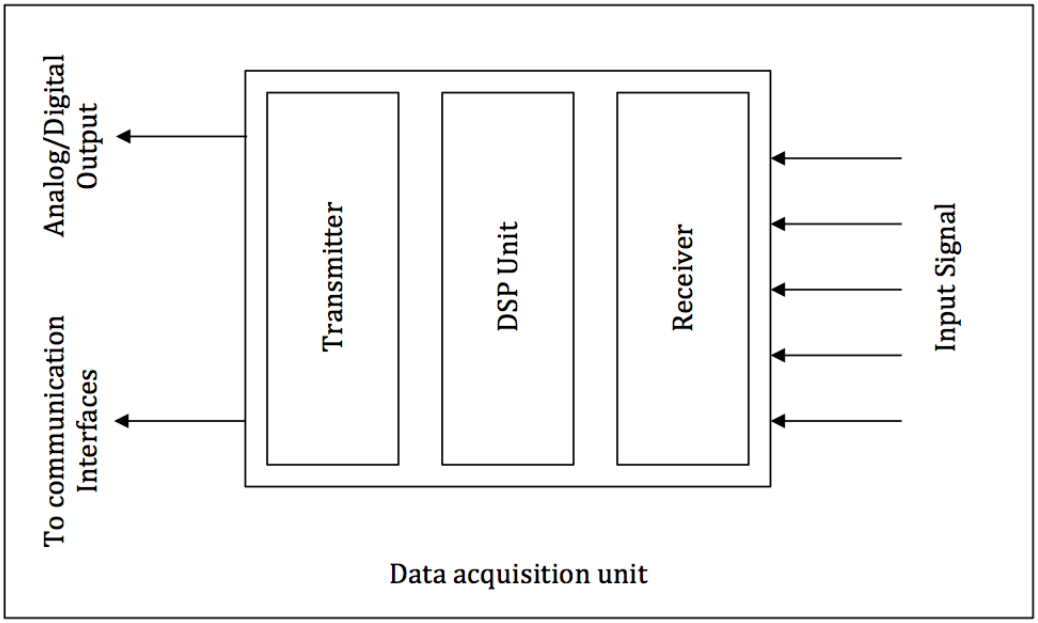


Fig. 1.1. Block diagram of a typical sensor

Capacitive: These accelerometers sense the change in the electrical capacitance between static condition and dynamic state with respect to acceleration.

Piezoelectric: These accelerometers use materials such as crystals, which generate electric potential from an applied stress, also called as the piezoelectric effect.

Piezoresistive: Accelerometers (strain gauge accelerometers) work by measuring the electrical resistance of a material when mechanical stress is applied.

Hall effect: Hall effect accelerometers measure voltage variations stemming from a change in the magnetic field around the accelerometer.

Magnetoresistive: Accelerometers work by measuring changes in resistance due to a magnetic field. The structure and function is similar to a Hall Effect accelerometer except that instead of measuring voltage, the magnetoresistive accelerometer measures resistance.

MEMS-based Accelerometers: MEMS technology is based on a number of tools and methodologies, which are used to form small structures with dimensions in the micrometer scale. The same technology is being utilized to manufacture state of the art MEMS-Based Accelerometers.

From industry to education, accelerometers have numerous applications. These applications range from triggering airbag deployments to the monitoring of nuclear reactors. There is a number of practical applications for accelerometers that are used to measure static acceleration (gravity), tilt of an object, dynamic acceleration, shock to an object, velocity, orientation and the vibration of an object.

Reasons for selecting MMA8452Q accelerometer:

1. It supports I2C communication (between accelerometer and the microcontroller)
2. It has two programmable interrupt pins for six interrupt sources:
    - It provides flexible output data that can be configured to be in either 8-bit or 12-bit
    - Motion/freefall detection is based on the configured threshold
    - It can detect single/double taps
    - It has the ability to detect the orientation in all 6 orientations
    - It has a built in high-pass filter along with user configurable cut off frequencies, which features transient detection
    - It has a built in auto-wake/sleep mode
3. It features dynamically selectable acceleration ranges of $\pm$2g/$\pm$4g/$\pm$8g
4. Its output data rates can be chosen from 1.56 Hz to 800 Hz depending on the signal resolution required by the application

1.6 Communication Protocol

The digital data from the sensors is usually provided in the serial form. We have some communication protocols, which are intended to use in these sensor applications, where the data can be of high resolution and frequencies, ranging from KHz to few MHz. Some of them are Inter Integrated Circuit protocol (I2C), Serial Peripheral Interface (SPI), Universal Asynchronous Receiver and Transmitter (UART), and Universal Serial Bus (USB) [13].

In this thesis I2C communication protocol is used, which is dependent on the clock frequency. All the communication and data transfer in this protocol is done with reference to the clock line. This I2C uses bidirectional open drain lines, Serial Data line (SDA) and Serial Clock line (SCL) pulled up with resistors. The typical voltages involved in this communication are 3.3V or 5V. The device that is controlling the other peripheral devices is called master and the devices connected to the master are called slaves. Each slave has its own address so that they can be invoked uniquely during operation. SPI is also similar to I2C but it has a chip select line to control the slaves connected to it. UART and USB communications are asynchronous communication protocols, where the data transfer and communication is done without a clock signal.

1.7 Proposed Approach

Triaxial signal based on scientific data features like SVM, SMA, and tilt angle are calculated and fed into the decision tree algorithm to obtain the real fall thresholds while eliminating false falls. Threshold values, determined from experimental verification, are used in the fall detection prototype, and whenever a fall is detected, an LED is turned on, and the event is logged. The proposed approach features high speed - low power from the use of low power and high speed embedded system processor. The algorithms used here also feature high speed to reach the processor decision in a timely manner.

2. NEUROSCIENCES AND NEUROSIGNALS

Prior to designing a reliable and effective fall detection system, it is necessary to study the neurological inputs and pathways of balance and natural fall prevention in humans. People monitor their environment by constantly adjusting their orientation with respect to movements. Two particular systems use external inputs to perform this task and anticipate the occurrence of a fall: vestibular system and somatosensory system.

2.1 Vestibular System

The vestibular system is in charge of engaging neurological pathways to provide perceptions of gravity and movement. The inner ear consists of a series of components that help transduce signals into electrical events. The membranes in the inner ear consist of three semicircular ducts (horizontal, anterior, and posterior), two otolith organs (saccule and utricle) and the cochlea, which is part of the auditory system. The semicircular ducts respond mainly to angular acceleration. A head turning movement induces movement of inner fluids that bend the cilia of hair cells. This causes the external input to convert into neurosignals. The otolith organs are located against the walls of the inner ear and they also influence the transmission of signals during head movements through the VIIIth nerve to the brainstem. The utricle organ has higher sensitivity when the head is upright, while the saccule is most sensitive when the head is in a horizontal position [14].

2.2 Somatosensory System

Somatosensory systems allow identifying the environment using physical touch. For instance, it helps process information about characteristics of the environment such as temperature and pain through neural stimulation. Also, the somatosensory system of proprioception causes awareness of body position through muscle and joint stimulation. This sensory information is transported and processed by somatosensory systems along various pathways based on the type of information that is being transported. For particular muscle contraction or proprioceptive information is carried along the column-medial lemniscal pathway [15].

2.3 Neurosignals

Different studies have shown how the proprioceptive system and muscle reactions influence body anticipation to a free fall in elderly subjects. Electromyography (EMG), a technique that helps study the muscle electrical activity, has helped to evaluate muscle activity during a fall. In Bisdorff's "EMG responses to free fall in elderly subjects and akinetic rigid patients" [16], EMG recordings in two normal subjects in response to randomly presented startling stimulus (fall) or non-startling stimulus (click) were analyzed. The subjects task was to dorsiflex the ankles in response to either stimulus. The fall induced startle occurred at about 100ms followed by the voluntary contraction at about 200ms. To assess the relative strength of the response, the rectified EMG areas were normalized in individual subjects by setting the strongest single activation found at an arbitrary level of 100%. The mean EMG strength was significantly larger in response to the startling stimulus (fall = 78.6 (SD 17.2)) than to the non-startling stimulus (click = 50.4 (SD 18.5); arbitrary % EMG units; $p = 0.0001$). It was concluded that in the case of a free fall it seems reasonable to assume that, in normal subjects, the vestibular system is important but, as the data suggest, patients with longstanding absence of vestibular function are capable of using other sensory sources to generate the response. Contact and proprioceptive

signals, particularly from the neck, have access to the brainstem at latencies only fractionally longer than vestibular ones and it could be important in detecting a fall and triggering motor responses [16]. Other conclusions include:

1. EMG responses in younger normal subjects occurred at: sternomastoid 54ms, abdominals 69ms, quadriceps 78ms, deltoid 80ms, and tibialis anterior 85ms. This pattern of muscle activation, which is not a simple ostrocaudal progression, may be temporally/spatially organized in the startle brainstem centers.

2. Voluntary tibialis EMG activation was earlier and stronger in response to a startling stimulus (fall) than in response to a nonstartling stimulus (sound). This suggests that the startle response can be regarded as a reticular mechanism enhancing motor responsiveness.

3. Elderly subjects showed similar activation sequences but delayed by about 20ms. This delay is more than that can be accounted for by slowing of central and peripheral motor conduction, therefore suggesting age dependent delay in central processing.

4. Avestibular patients had normal latencies indicating that the free fall startle can be elicited by non-vestibular inputs.

5. Latencies in patients with idiopathic Parkinsons disease were normal whereas responses were earlier in patients with multiple system atrophy (MSA) and delayed or absent in patients with Steele-Richardson-Olszewski (SRO) syndrome. The findings in this patient group suggest

    - Lack of dopaminergic influence on the timing of the startle response.
    - Concurrent cerebellar involvement in MSA may cause startle disinhibition.
    - Extensive reticular damage in SRO severely interferes with the response to free-fall.

2.4 Experiment Protocol

Based on the results of Bisdorffs [16] shown in Section 2.3, and after obtaining St. Vincents Hospital Internal Review Board (IRB) approval for experimentation with human subjects (R2010-138), the following protocol for experimentation was developed:

2.4.1 Test A

Two rounds of testing were performed. For the first round, test A, six healthy volunteers (3 male, 3 female) between the ages of 21 and 35 years old were recruited. Five wireless sensors were positioned in five different places as shown in Figure 2.1. Each person performed different fall types including forward, backward, and sideways, falling while transitioning from chair to standing position. Non-fall data was also recorded that included walking and bending.

2.4.2 Test B

For the second round of experiments ten healthy volunteers (5 male, 5 female) between ages of 21 and 40 years old were recruited. Similar to the first experimental set, the subjects performed the same fall types but this time the data was collected only from sensors S1, S2, and S3 for simplicity and high noise in the activity of lower extremities. The network camera was also used in test B to record all the activities, falls and no falls, of every subject.

The sensors used for Test B set of experiments are from Freescale semiconductors, called ZSTAR3, shown in Figure 2.2. The ZSTAR3 has MMA7361LT low power capacitive accelerometer on it. Three ZSTAR3 sensors are placed on the patients body and the data is acquired using a wireless USB stick. The wireless communication is done at 2.4 GHz Radio Frequency. ZSTAR3 triaxial accelerometer sensor has a selectable data rate of 30, 60 or 120Hz. The wireless range of these sensors is up to

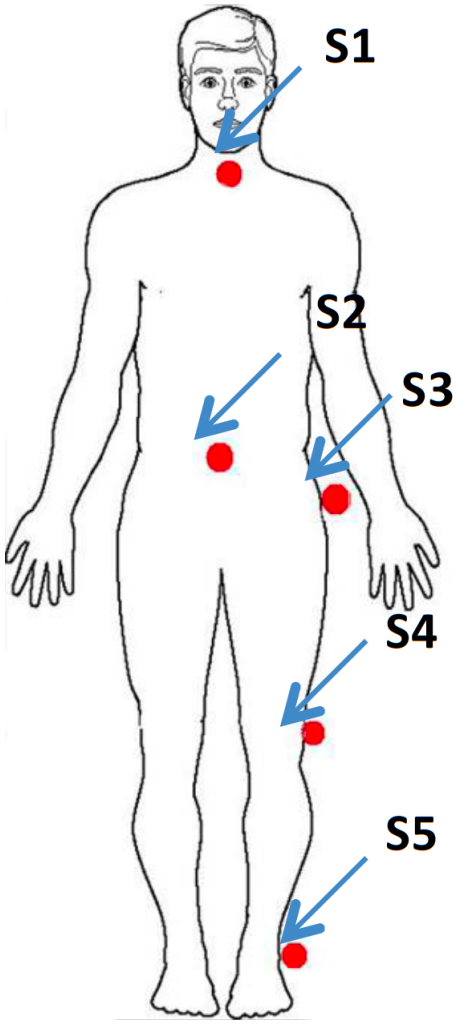


Fig. 2.1. Sensors placement

20 meters. It consumes 1.8 to 3.9 mA of current during normal mode of operation. A coin sized CR2032 3V battery powers the sensors.

During the data acquisition from accelerometer sensors, all the falls and non-fall events are recorded using an IP camera to have a log of fall time so that they can be used while processing the data for thresholds using decision trees. The IP camera used in this project is IQinVision IQEYE2803A4, and is shown in Figure 2.3. This IPcamera is set up using a File Transfer Protocol Server (FTP) and FTP client. The filezilla software is set up in such a way that the camera data is transmitted to an

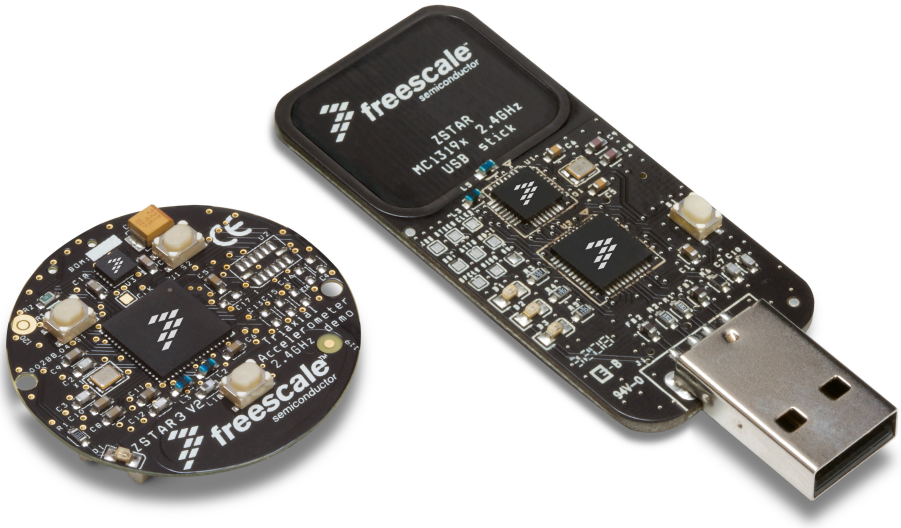


Fig. 2.2. ZSTAR3 sensor and USB stick

external hard drive connected to a computer through Wi-Fi. The camera data is obtained in the form of sequential images with a time stamp on them.

Fig. 2.3. IPcamera used during experiments

2.4.3 Test C

For the third round of experiments six healthy volunteers (4 male, 2 female) between ages of 21 and 40 years old were recruited. Similar to the second experimental setup, the subjects performed the same fall types but this time hardware prototype is used. The details of this hardware is described in Chapter 3. Figure 2.4 illustrates the placement of sensors and the processing unit.

Reasons for opting the new Hardware prototype:

- The new fall detection unit is an integrated wired sensor system because; wireless sensors are prone to high power usage when compared to wired.
- Wireless sensors are more vulnerable to signal interference and they need to have individual power supplies.
- Wireless sensors are not easy to carry, as they are not held together. Due to this reason they need to be calibrated each time when the patient uses it.
- The new hardware prototype comes with a vest, to which all the three accelerometers are sewed and the processing unit is also attached to it.
- These accelerometers have long cables such that it can be adjusted on the vest for people of different heights.
- The processing unit handles the fall detection when thresholds are met and also it has a Bluetooth module to transmit the data to any Bluetooth enabled device wirelessly.

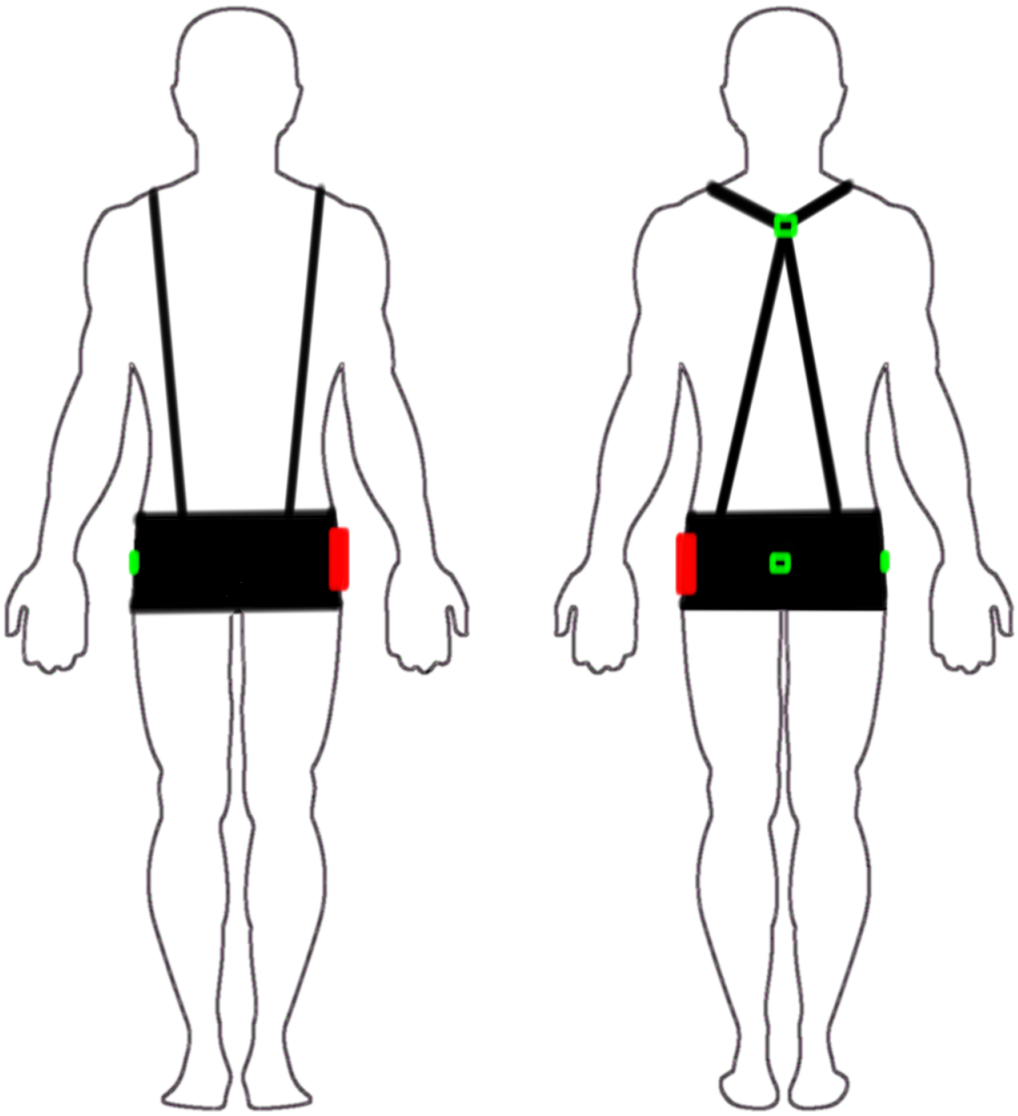


Fig. 2.4. Front and back view of Proposed hardware prototype. Green blocks represent the accelerometers and the red represents the processing unit

3. HARDWARE DESIGN

This chapter provides the information about the hardware design and integration of sensors into an embedded system. The embedded system using I2C communication between the micro controller and the sensors is detailed. The serial data from the micro controller is transmitted using a Bluetooth module.

3.1 The Embedded System

The integrated system consists of triaxial accelerometers, I2C multiplexer, micro controller and Bluetooth module. The following block diagram Figure 3.1 gives the basic idea of the integrated hardware and the types of communication between the components. The ATmega328 is the Major control unit, to which everything is integrated. The accelerometers are connected to this control unit through an I2C multiplexer such that the Microprocessor can distinguish between them even though they have the same addresses. The information from the accelerometers is processed and the data is wirelessly transmitted to mobile devices or laptop using the Bluetooth module connected to it.

3.1.1 Arduino

Arduino UNO microcontroller unit used in this project is an open source hardware. It has ATmega328 micro controller on it. It comes with a total of 14 digital input/output pins and 6 analog inputs. Out of the 14 digital input/output pins, 6 can be used as Pulse Width Modulation (PWM) outputs. It has a 16 MHz ceramic oscillator onboard.

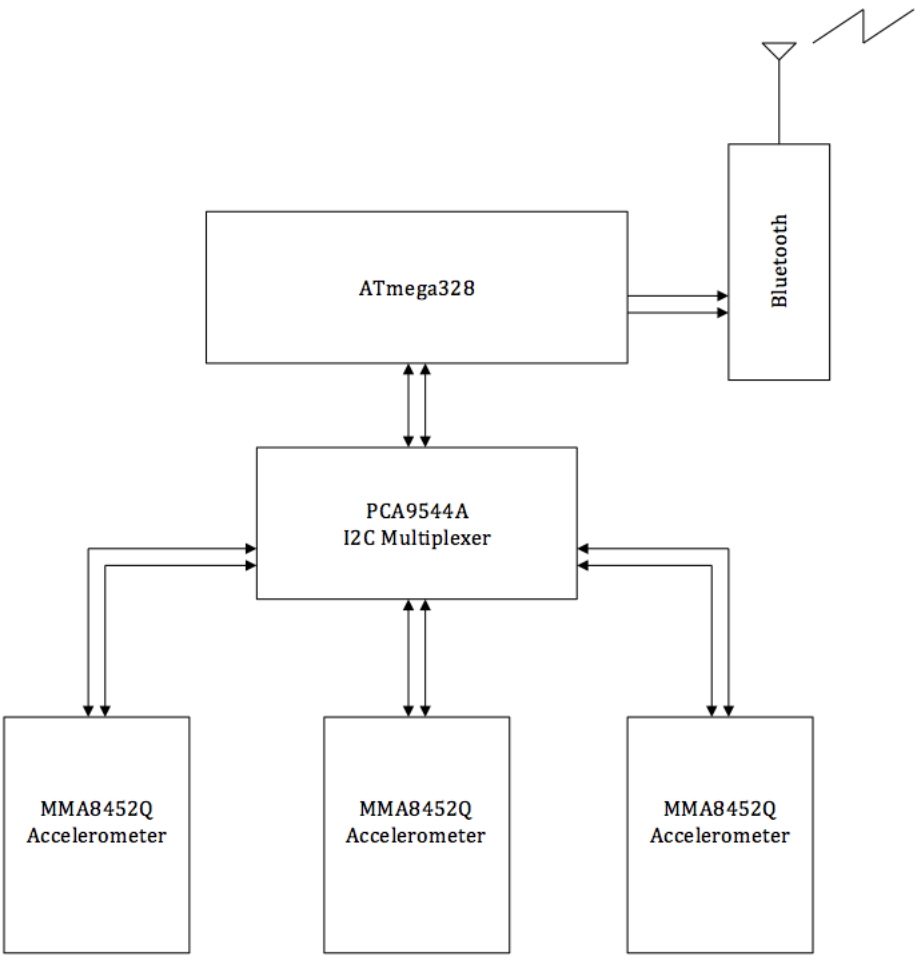


Fig. 3.1. Block diagram of the integrated sensor system

The ATmega328 has 32KB Flash Memory, 2KB SRAM and 1KB EEPROM. The Arduino UNO board operating voltage is 5V and it has an onboard voltage regulator which can take up to a maximum of 20V from the supplied power jack. The board can be programmed using the USB port available and it can also be powered using the same port. The power jack provided can be used to run the Arduino when it is not used with the USB. There is an ICSP header for debugging and also a reset push button. In this project we are using the analog pins A4 and A5 pins to connect the SDA and SCK of the I2C multiplexer. Digital pin D2 is used as logical low interrupt for the multiplexer. The

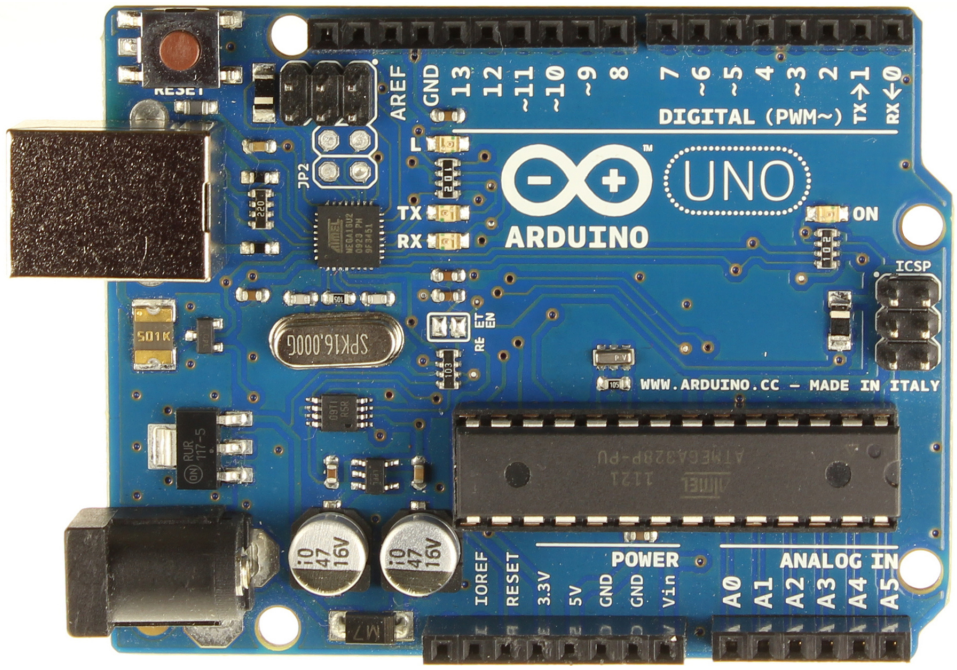


Fig. 3.2. Arduino UNO board

Bluetooth module is connected to Rx and Tx pins of the Arduino so that the serial communication can be done wirelessly. Figure 3.2 details the Arduino UNO prototyping board and Figure 3.3 shows the pin diagram of ATmega328.

3.1.2 MMA8452Q Accelerometer

The MMA8452Q is a 3-Axis, smart, low-power, capacitive micromachined Digital Accelerometer from Freescale semiconductors with 12 bits of resolution [17]. It is packed with two interrupt pins, which can be used to invoke the inbuilt flexible user programmable options and embedded functions. Those embedded interrupt functions allow for overall power savings relieving the host processor from continuously polling data.

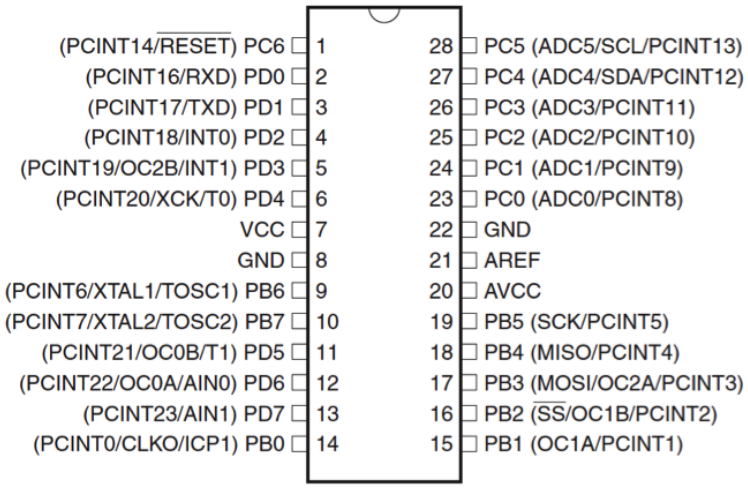


Fig. 3.3. ATmega328 pin diagram

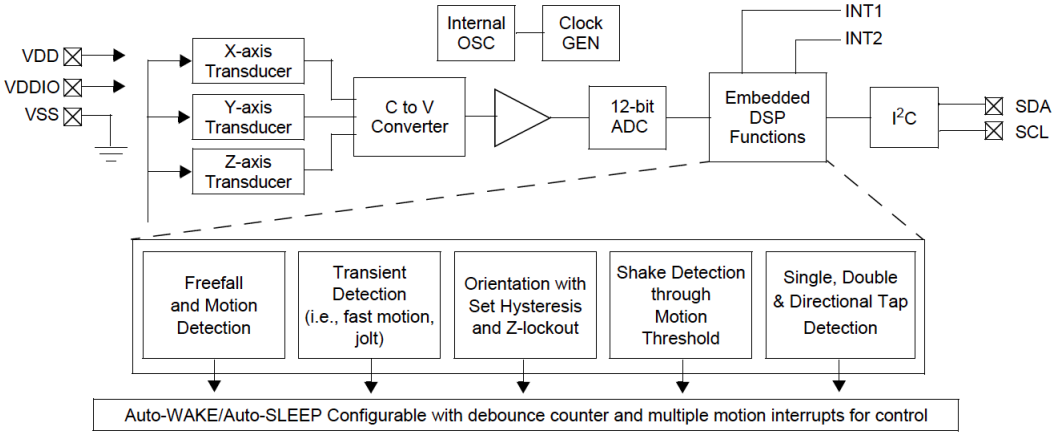


Fig. 3.4. MMA8452Q system architecture

The MMA8452Q has user selectable full scales of ±2g/±4g/±8g with high-pass filtered data available for real-time applications. The communication is done using the I2C digital output interface. The MMA8452Q accelerometer has 42 configurable registers, which can be used based on the application. The acceleration data of the X, Y and Z-axes are stored as 2's complements of 12-bit numbers of the 6 registers from 0x01 to 0x06. Some of the features are motion

freefall detection, tap and pulse detection, orientation, high pass filtering, Auto-sleep and wake up. The Accelerometer is small enough for patients to wear. The MMA8452Q operating at 800Hz consumes 165μA current, making it a perfect choice for this application. Figure 3.4 and Figure 3.5 give the block and circuit diagrams that detail the internal architecture and pin connections of accelerometer.

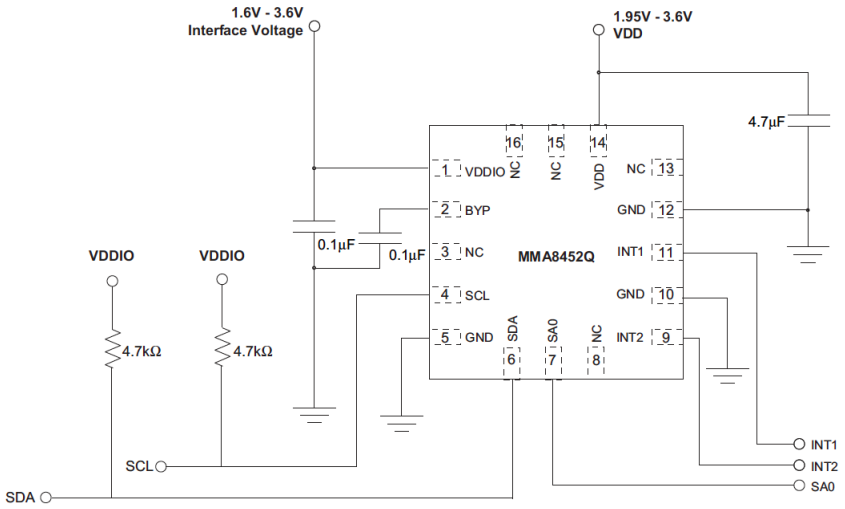


Fig. 3.5. MMA8452Q pin connections

3.1.3 I2C Multiplexer for Accelerometers

The accelerometers used in design of this prototype are MMA8452Q from Freescale semiconductors. The data from the accelerometers are read using I2C communication. I2C communication is done based on the address of the slave units connected to the master unit. All the 3 accelerometers that are used in the design have the same address 0x2A. PCA9544A 4-channel I2C-bus multiplexer, a quad bidirectional-translating switch, is used to regulate the switching between the three accelerometers, where one SCL/SDA pair can be selected at a time [18]. The PCA9544A provides four interrupt inputs and one open drain

interrupt output. Whenever any device generates an interrupt, it is detected by the multiplexer and the interrupt output is driven low. Out of the four channels available, three were used to do the communication with the accelerometers. The multiplexer has a unique address of 0x70 while the SDA and SCL are connected to A4 and A5 of the Arduino. Figure 3.6 gives the pin diagram and Figure 3.7 is the sample application of PCA9544A I2C multiplexer.

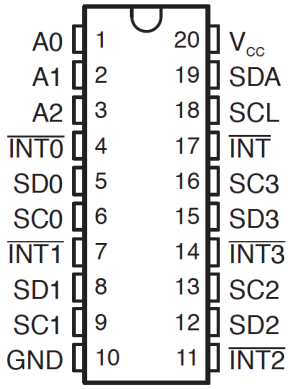


Fig. 3.6. PCA9544A pin diagram

3.1.4 Bluetooth

The Bluetooth module used in this prototype is a factory configured serial data transmission board. It has Vcc, Tx, Rx, and ground pins of which the Tx of the Bluetooth is connected to Rx of Arduino, and the Rx of Bluetooth is connected to the Tx of Arduino in order to transfer the data wirelessly. The Bluetooth module is configured to 9600 Baud rate as a default setting. It can operate at a range of up to 30ft and voltage range from 3.3 to 5 V.

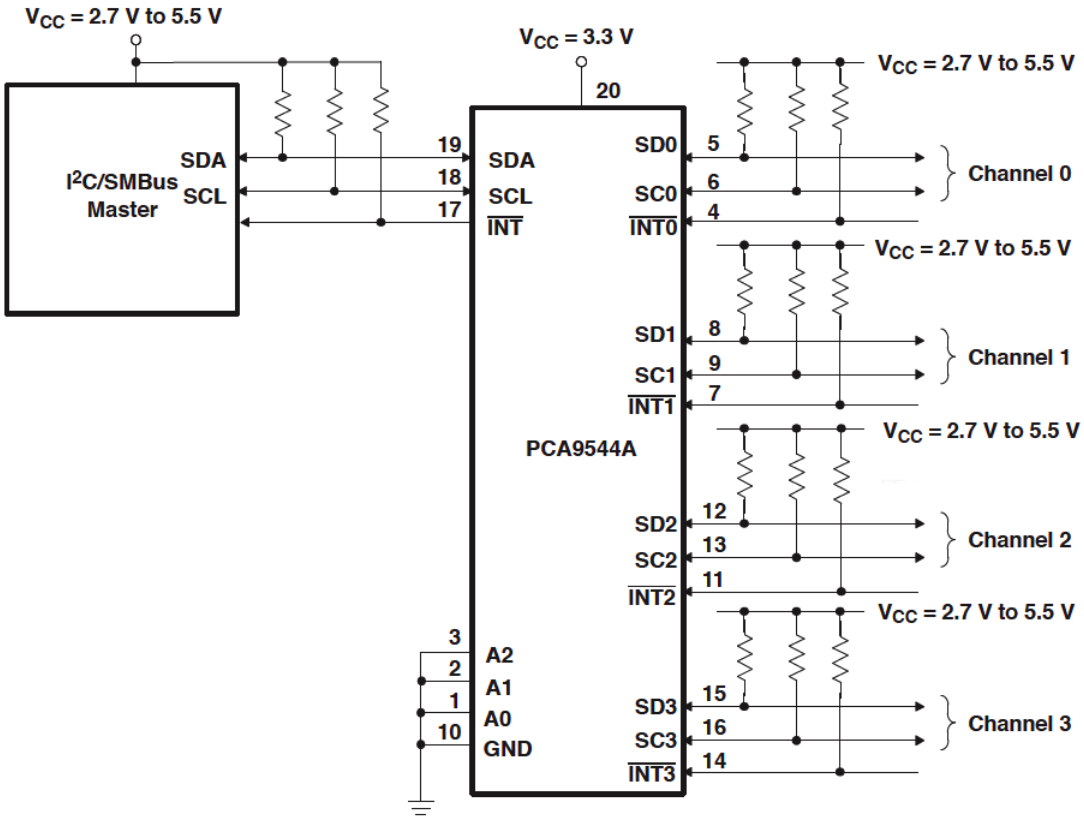


Fig. 3.7. Application of PCA9544A I2C multiplexer

Fig. 3.8. Bluetooth module

3.2 Inter-Integrated Circuit Communication- I2C

Inter-Integrated Circuit is a bidirectional two-wire interface synchronous communication protocol. It requires two bus lines, Serial Data and Serial Clock.

Each device connected to this bus is software addressable by a unique address. I2C bus is a multi-master bus where more than one integrated circuit is capable of initiating a data transfer can be connected to it, which allows masters to functions as transmitters or receivers. I2C communication is highly immune to noise, has wide supply voltage range that consumes very low current.

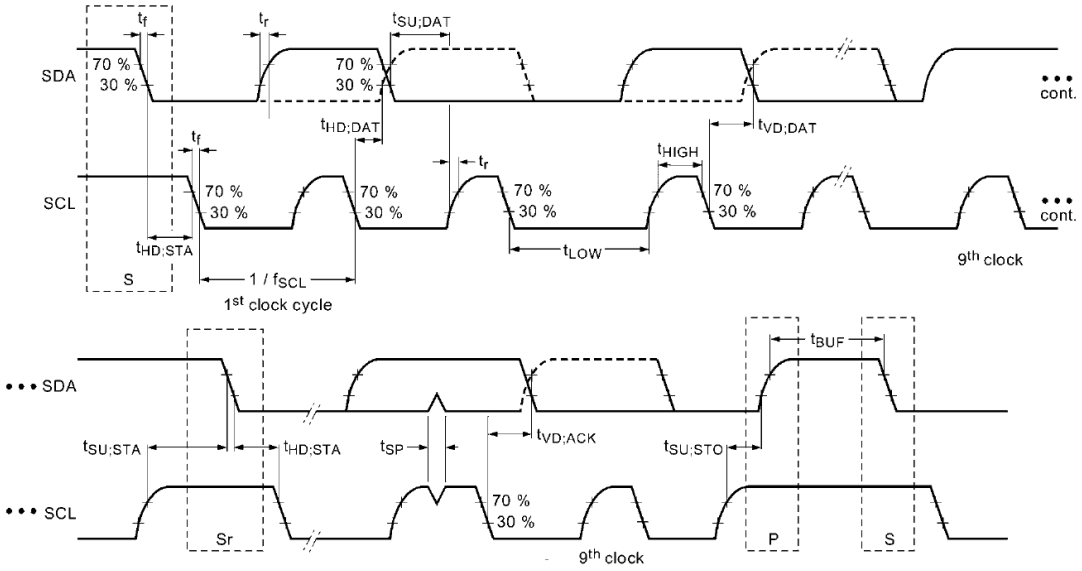


Fig. 3.9. I2C timing diagram

In this thesis, the microprocessor acts as a master and the three triaxial accelerometers act as slaves. Both the bi-directional lines, SDA and SCL are connected to a positive supply voltage via 4.7KΩ pull up resistors. Data transfer rate on the I2C bus can range from 100Kbits/s to 3.4 Mbits/s based on the application modes. A data START condition is observed when a HIGH to LOW transition on the SDA while SCL is HIGH. A LOW to HIGH transition on the SDA line while SCL is HIGH defines a STOP condition. These START and STOP conditions are always generated by the master. Once the START condition is initiated, the bus is considered as busy until a STOP condition is reached. Figure 3.9 shows the signal integrity timing diagrams, including the START and STOP bits.

3.3 UART Communication

The Universal Asynchronous Receiver/Transmitter communication is used to transmit the data from three accelerometers to the mobile device using the Bluetooth module. Unlike I2C, UART is an asynchronous communication protocol (No clock required). Baud rate for the Bluetooth module used in this thesis is set to 115200.

3.4 Hardware Programming

- The wire and math libraries are included for the I2C communication and trigonometric functions respectively.
- Initially to begin the I2C communication with the accelerometers, the contents of 0x0D register is read using the readRegister user defined function.
- This readRegister function invokes the Wire.beginTransmission function of the Arduino library, which begins a transmission to the I2C slave device with the address 0x1D.
- The Wire.write(0x0D) function writes the data from the accelerometer in response to a request from the ATmega328.
- The Wire.endTransmission(false) command is used not to send a STOP condition to the Wire.beginTransmission such that the I2C bus will not be released yet. This prevents another master device from transmitting between messages. This allows one master device to send multiple transmissions while in control.
- Wire.requestFrom(address, quantity) is used by the master to request bytes from a slave device.
- Wire.read( ) reads a byte that was transmitted from a slave device to a master after a call to requestFrom() was transmitted from a master to slave.

- The measured acceleration data of the MMA8452Q is stored in OUT_X_MSB, OUT_X_LSB, OUT_Y_MSB, OUT_Y_LSB, OUT_Z_MSB, and OUT_Z_LSB registers as 2s complement 12-bit numbers. The most significant 8-bits of each axis are stored in OUT_X (Y, Z)_MSB
- The MMA8452Q has an internal ADC that can sample, convert and return the sensor data when requested.
- The 8-bit command transmission begins on the falling edge of SCL.
- The transaction on the I2C bus starts with a START condition signal. After START condition has been transmitted by the master (ATmega328), the I2C bus is considered as busy.
- The next byte of data transmitted after START contains the slave address in the first 7 bits, and the eighth bit is reserved to indicate whether the master is receiving data or transmitting data.
- The MMA8452Q is set to operate at 800 Hz (Maximum available) such that it can transmit 84 samples per second when 115200 baud rate is used.
- Signal features SVM, SMA and Tilt angle are calculated and the thresholds are set such that whenever there is a fall occurrence, the LED pin connected to the 12th pin of Arduino is turned on and the event is logged on the PC.
- The Arduino UNO board is programmed using the Arduino software and the code is included in the appendix section.

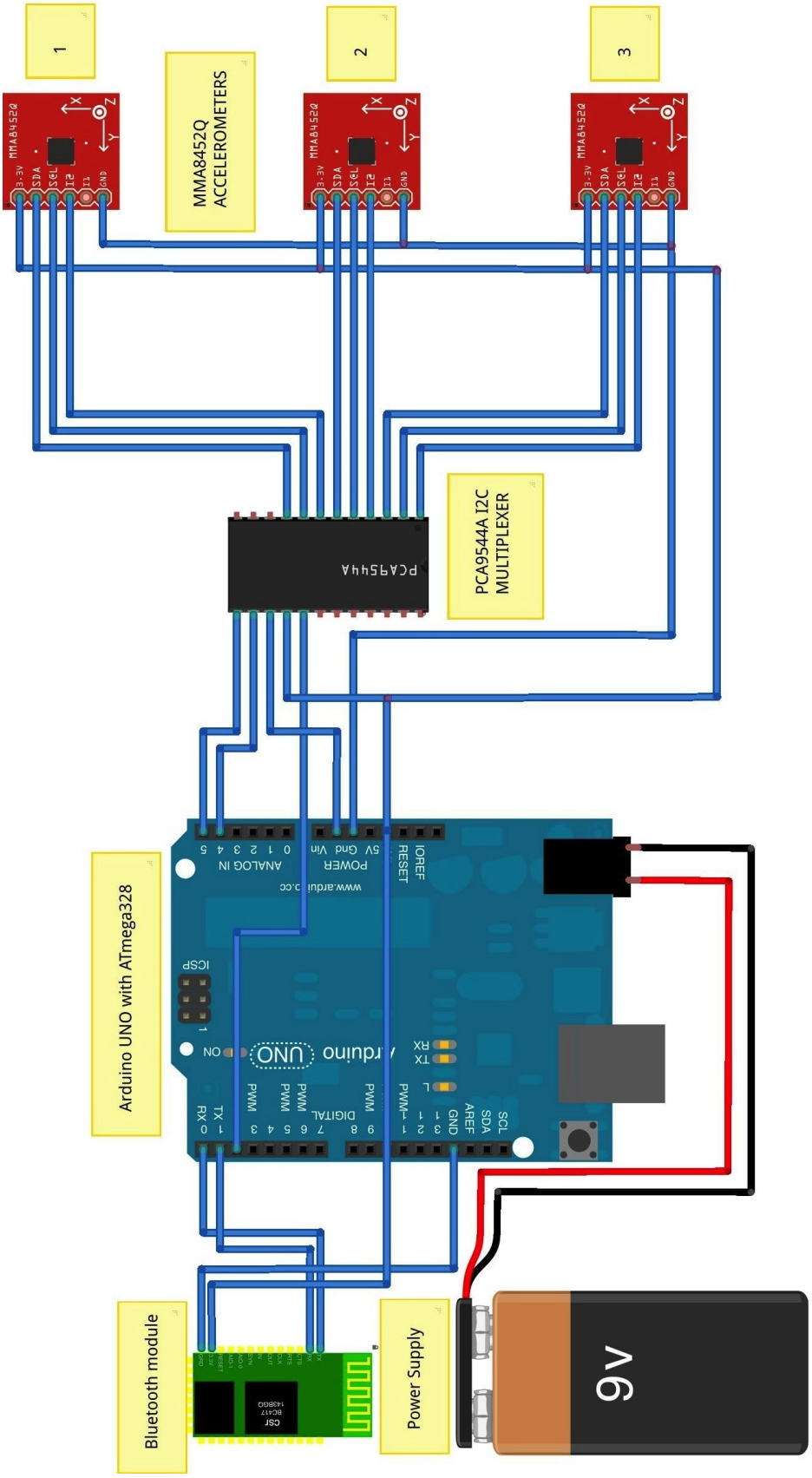


Fig. 3.10. Circuit connections for the hardware prototype

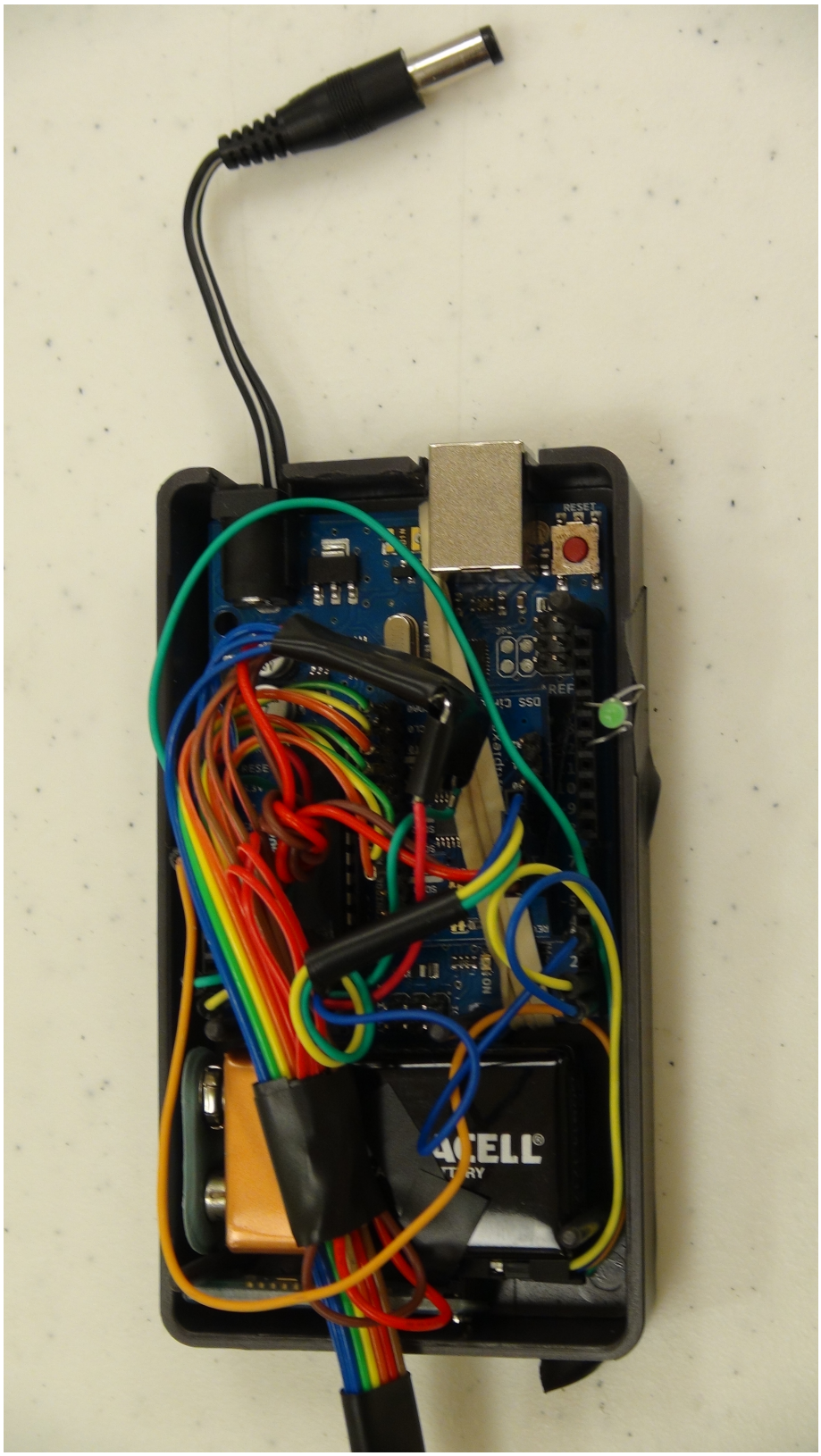


Fig. 3.11. Image illustrating hardware prototype enclosed in a plastic box

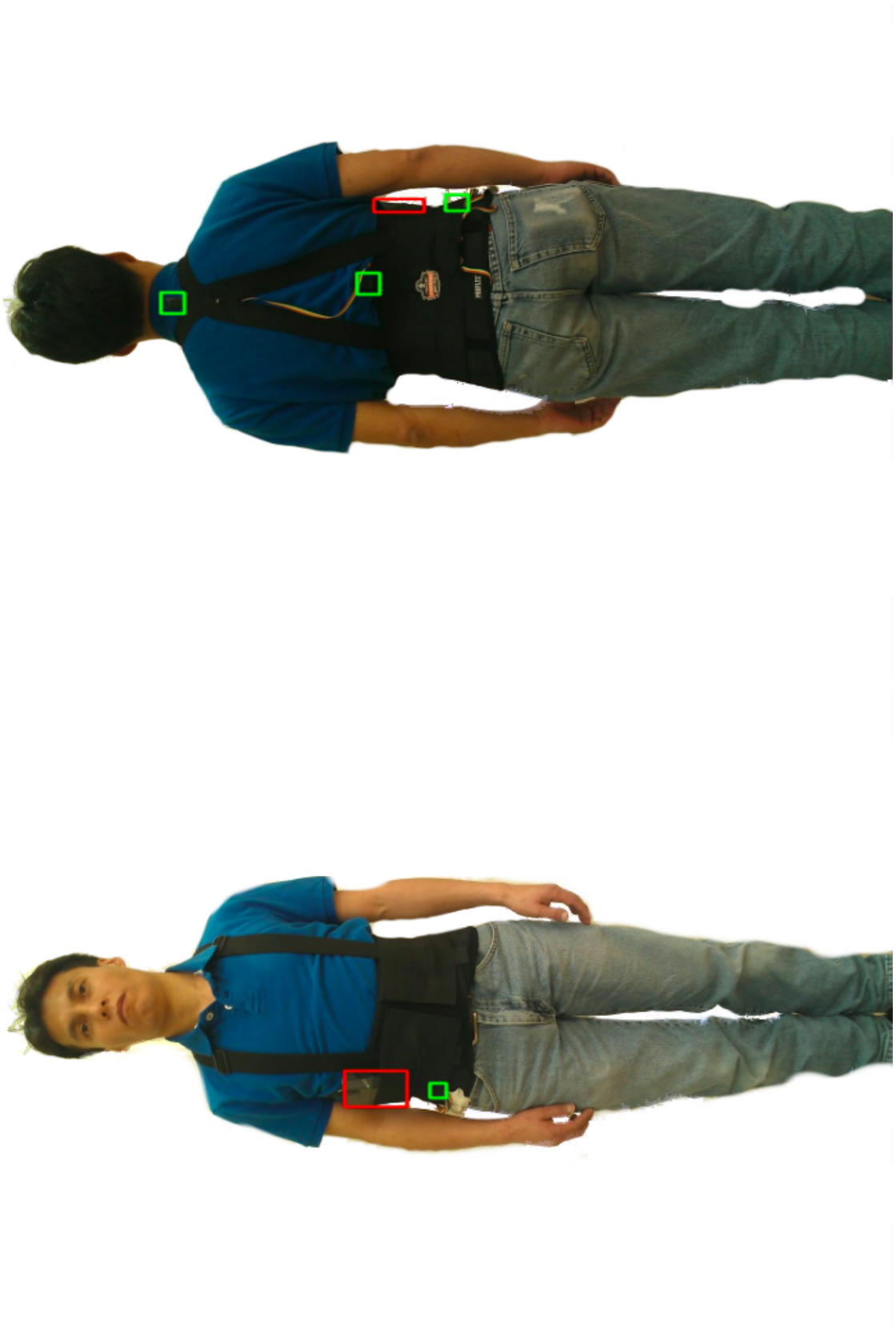


Fig. 3.12. Subject demonstrating the fall detection unit

4. PATTERN RECOGNITION AND DATA ACQUISITION

In this study the two preprocessing steps performed by Karantonis et al. [19]are used. The first step is median filtering and the second step is low pass filtering. The low pass signal filtering is considered as an estimation of the gravitational acceleration (GA), and the median filtering is an estimation of the body acceleration (BA).

4.1 Feature Extraction Indices SVM, SMA, Tilt Angle

In this study we used the second algorithm presented by Karatonis et al. [19]. The algorithm is based on the assumption that a fall is a signal of extreme impact. The degree of movement intensity is known as signal vector magnitude (SVM) and it is derived from the BA component as follows:

$$SVM[i] = \sqrt{x^2{}_{BA}\,[i] + y^2{}_{BA}\,[i] + z^2{}_{BA}\,[i]} \tag{4.1}$$

where $x_{BA}\,[i]$ is the i^{th} sample of the BA component along the axis samples (similarly for $y_{BA}\,[i]$ and $z_{BA}\,[i]$). Comparing the SVM with a threshold helps determine the fall event. In order to measure the intensity of the activity and distinguish between rest and movement, the signal magnitude area (SMA) is calculated. SMA is the sum of the integrals of the three acceleration signal magnitudes and it is also calculated using the BA component as shown:

$$SMA\,[i] = \frac{1}{T} \sum_{j=i}^{j=i-T} \left(|x_{BA}\,[j]| + |y_{BA}\,[j]| + |z_{BA}\,[j]|\right) \tag{4.2}$$

where $x_{BA}[i]$, $y_{BA}[i]$, and $z_{BA}[i]$ are the BA components of the x, y , and z axis signals and T is the sampling period. Using the GA component of the signal helps determine the postural orientation of the subject wearing the accelerometers. The derivation of tilt angle can be achieved using the GA component along the z axis as

$$\Phi[i] = \cos^{-1}\left(\frac{z_{GA}[i]}{\sqrt{x^2{}_{GA}[i] + y^2{}_{GA}[i] + z^2{}_{GA}[i]}}\right) \tag{4.3}$$

where $x_{GA}[i]$ is the i^{th} sample of the GA component along the axis samples (similarly for $y_{GA}[i]$ and $z_{GA}[i]$)

4.2 Threshold Information

Using data collected after Test A and Test B described in Chapter 2 section 4 and video recordings of the fall, image processing techniques were used to classify the accelerometer data as fall and no fall events based on the body inclination. The images were processed in order to determine the moment in which body inclination was between 15 and 60 degrees with respect to the vertical axis. Using that moment in time where the image reached the range, the accelerometer data was then classified as fall or no fall (0 for no fall and 1 for fall). Matrices containing the classification array of zeros and ones, and arrays of SMA values, SVM, and Tilt angles for the three sensors were created and used to find thresholds using a decision tree model.

4.3 Decision Tree Model

Decision trees are pattern recognition tools that provide weighted solutions to a classification problem with output classes such as fall/no fall in our case. Decision Trees are constructed from training data sets in which each data point

contains an input vector along with a target value. The target value, either a 1 or a 0, represents the class to which the data belongs. The software Rattle, a sub-package of R was used for the purpose of training and calculating the fall/no fall threshold values. These thresholds are later coded using if-then statements and later stated on unseen data points as the prediction is compared with true classes. Figure 4.1 shows a sample DT as a flowchart with rules. Two parameters in Rattle are adjusted to modify the output: complexity cost and loss matrix. The complexity cost is a number between 0 and 0.0001 that adjusts the size of the tree. The larger the complexity costs the simple decision tree containing fewer nodes. The loss matrix is a comparative misclassification cost used to make fall or no fall class almost pure.

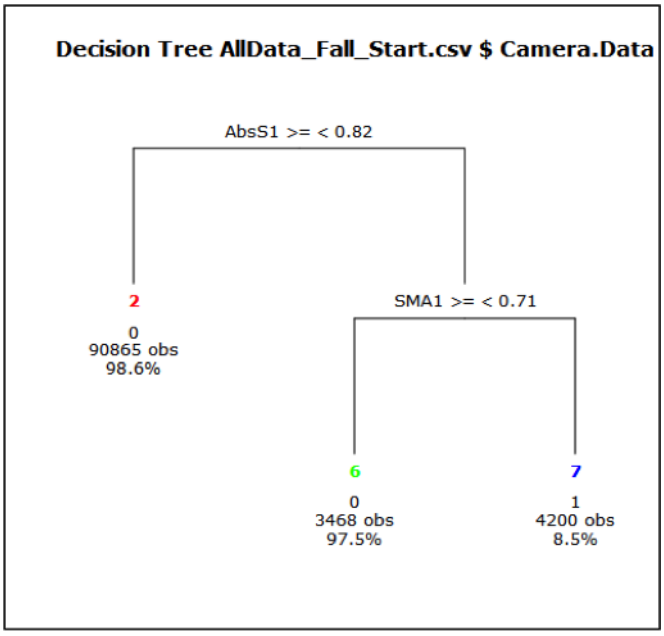


Fig. 4.1. Example decision tree

4.4 Falls and Non Falls Setups

Using the new hardware prototype of the integrated sensor system, data was collected from 6 subjects (4 male and 2 female). The age, height and weight of all subjects were documented. All the six subjects were asked to wear the vest, to which the sensors and the processing unit were attached. Seven different activities, which imitate both falls and non-falls, are asked to perform:

- **Frontal fall:** Subjects were asked to take two laps of normal walking around the mattress and to imitate a frontal fall on the mattress.
- **Side fall:** Subjects were asked to take one lap and take a side fall on the mattress.
- **Back fall:** This fall was taken without walking, but asked to fall down backwards.
- **Chair fall:** Before the data acquisition, the subject will be sitting in a chair and asked to imitate a chair fall while standing up, and the data is collected.
- **Sit normal in a chair:** This involves the subject sitting in a chair normally.
- **Sit sudden in a chair:** In this, the subject is asked to sit suddenly and it should be considered as a non-fall by the detection unit.
- **Tripping:** Subjects are asked to walk normally for some time, then imitate a trip near a window but prevent themselves from falling. This should be detected as a no fall by the hardware unit.
- **Lay normal on a bed:** Subjects were asked to walk around and lay normally on a bed.
- **Lay suddenly on a bed:** This is similar to normal laying but the subject will be doing it with a sudden movement.

Out of these 7 activities, the first 4 are the real falls and the later 3 are non-falls. Accuracy is determined based on the true positives, true negatives, false positives and false negatives of the fall detection. The following figures illustrate some of the fall types and also give an idea of the test setup.

4.5 Data Acquisition Using Bluetooth Enabled Laptop

The serial data transmitted by Bluetooth module connected to the processing unit can be saved on any computer that has Bluetooth capability. The pairing password for the Bluetooth module is 1234 by default. The baud rate is also set to 9600 as factory default. As we need to transfer our serial data at 115200 baud rate, it can changed by sending some AT commands to it. AT+BAUD8 command will change the baud rate from 9600 to 115200. Serial data from the Bluetooth can be saved on a computer in command separated values file version (csv) using MATLAB as shown in Figure 4.6. The Bluetooth device can be identified and used to save data using the command:
s=serial('/dev/tty.BTUART-DevB'); for Mac
s=serial('COM4'); for Windows
set(s, 'BaudRate', 115200); is used to set the baud rate of the port.
datestr(now,'HH,MM,SS,FFF'); is used to store the data with a time stamp in Hours:minutes:seconds:milliseconds format.

The falls are detected in real time using the thresholds set on the signal features. During the experimentation process, the data is transmitted in real time to a Bluetooth enabled device to verify the accuracy of hardware prototype.

Figures 4.2 to 4.5 shows four diffrent types of falls visually and Figure 4.7 illustrates a frontal fall graphically with the SVM from 3 accelerometers.

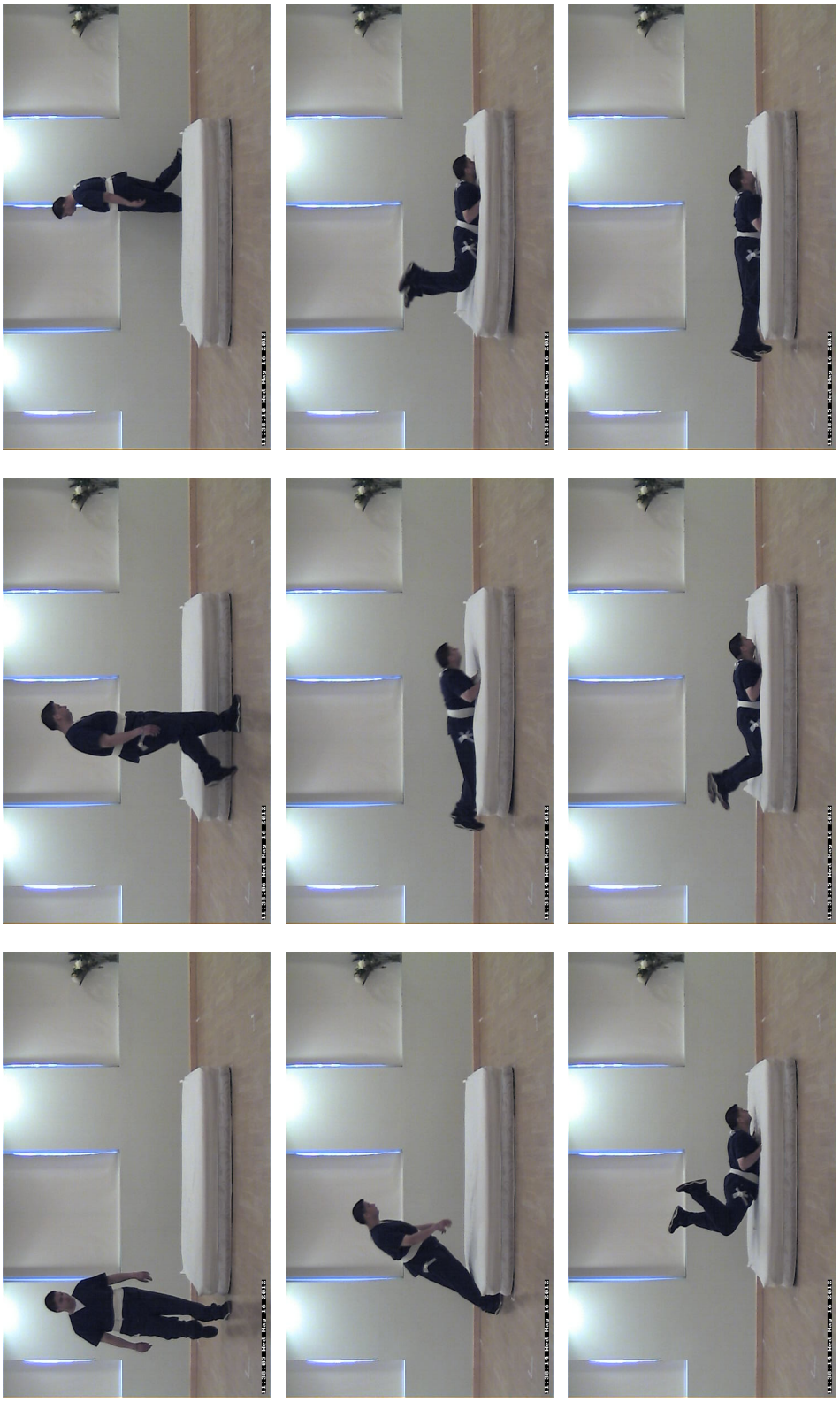


Fig. 4.2. Frontal fall demonstration

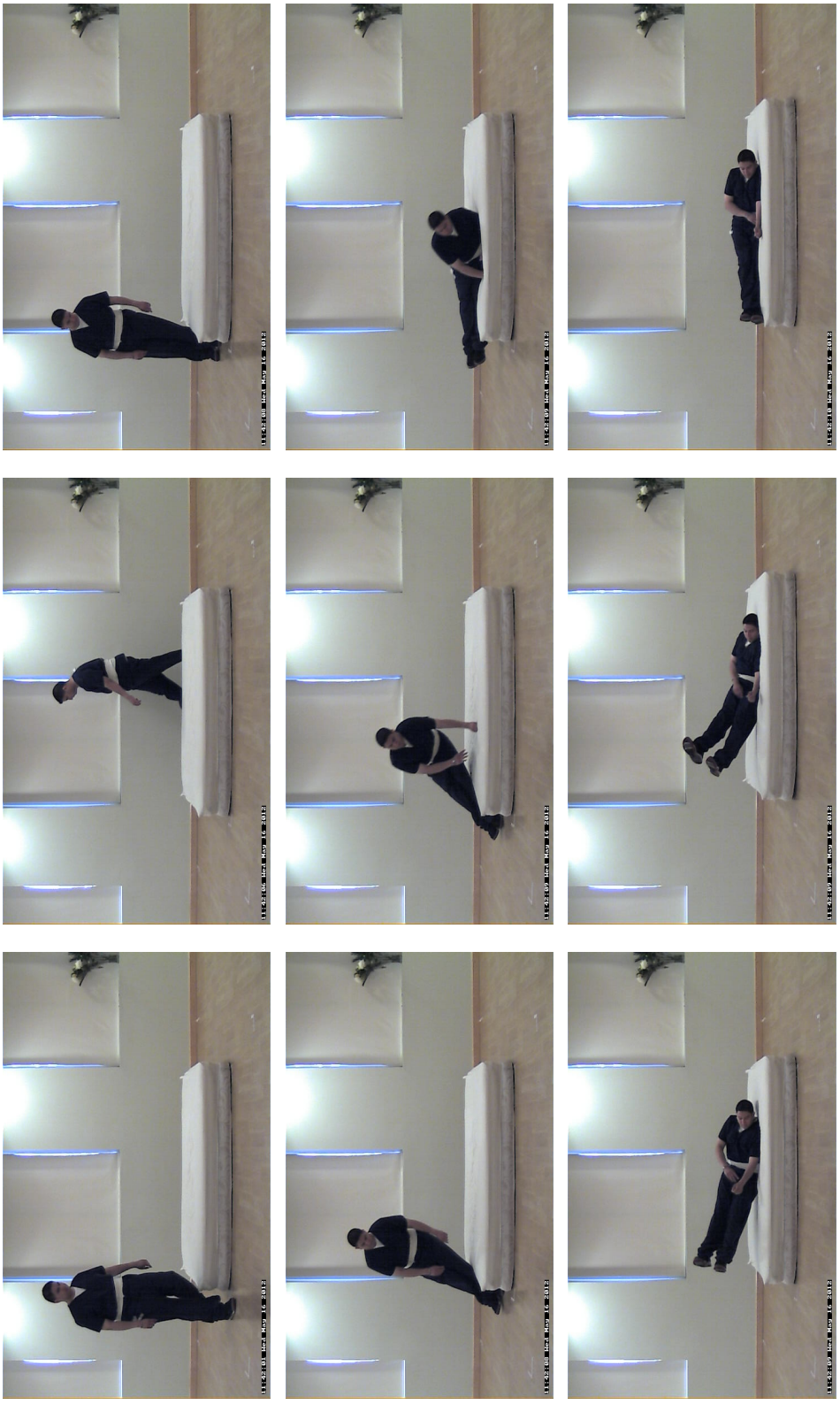


Fig. 4.3. Sideways fall demonstration

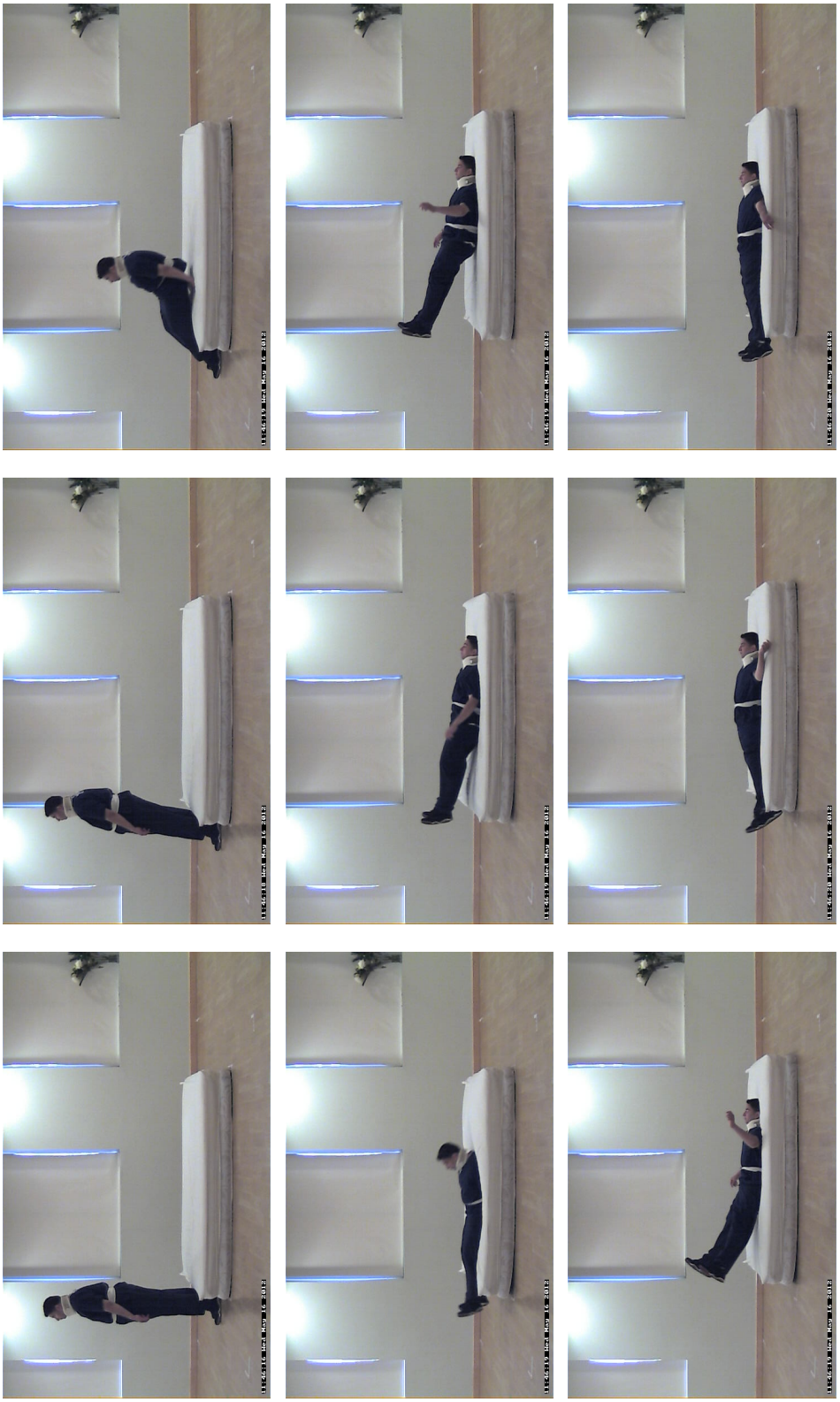
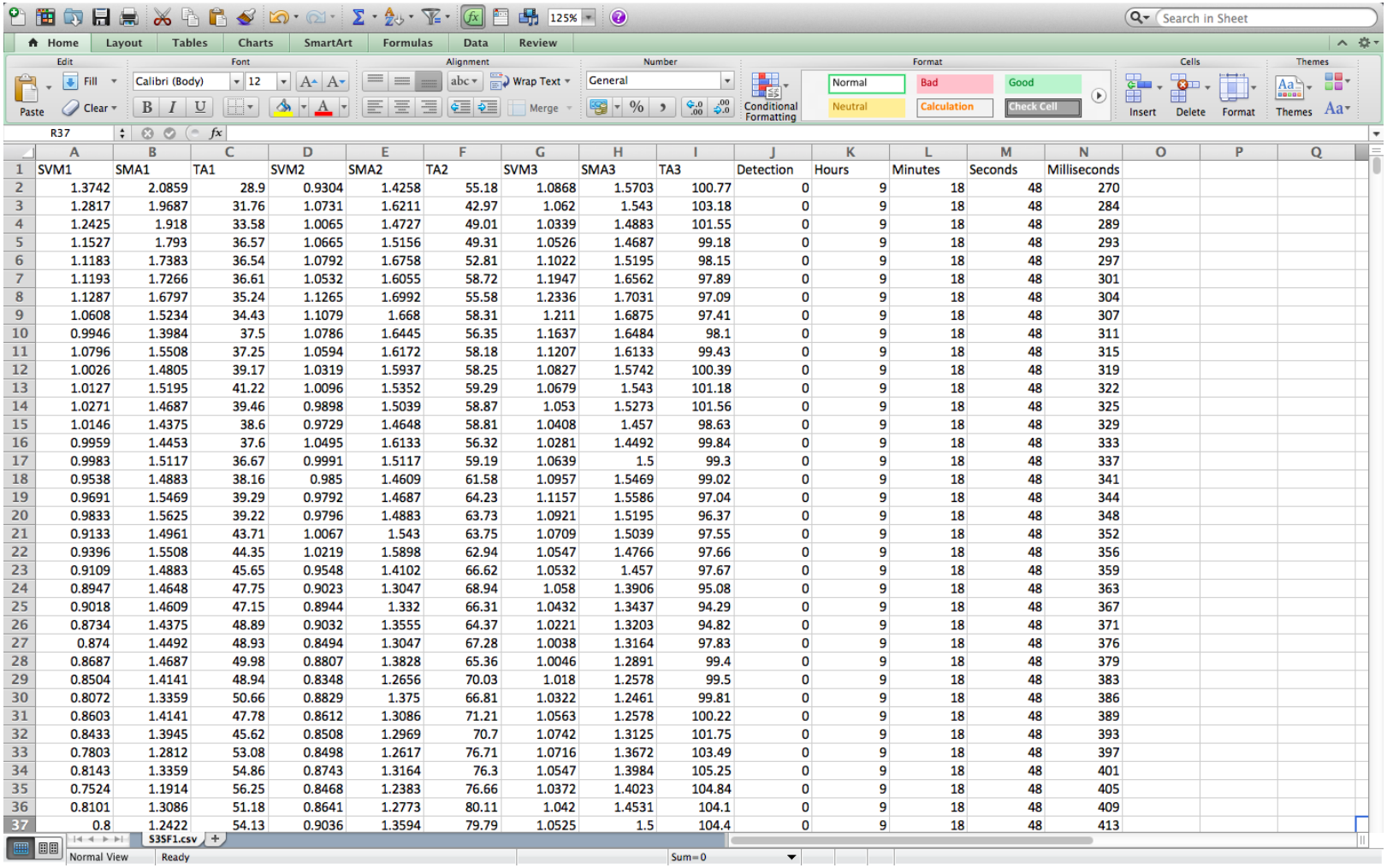


Fig. 4.4. Backwards fall demonstration

Fig. 4.5. Falling from a chair demonstration



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	SVM1	SMA1	TA1	SVM2	SMA2	TA2	SVM3	SMA3	TA3	Detection	Hours	Minutes	Seconds	Milliseconds			
2	1.3742	2.0859	28.9	0.9304	1.4258	55.18	1.0868	1.5703	100.77	0	9	18	48	270			
3	1.2817	1.9687	31.76	1.0731	1.6211	42.97	1.062	1.543	103.18	0	9	18	48	284			
4	1.2425	1.918	33.58	1.0065	1.4727	49.01	1.0339	1.4883	101.55	0	9	18	48	289			
5	1.1527	1.793	36.57	1.0665	1.5156	49.31	1.0526	1.4687	99.18	0	9	18	48	293			
6	1.1183	1.7383	36.54	1.0792	1.6758	52.81	1.1022	1.5195	98.15	0	9	18	48	297			
7	1.1193	1.7266	36.61	1.0532	1.6055	58.72	1.1947	1.6562	97.89	0	9	18	48	301			
8	1.1287	1.6797	35.24	1.1265	1.6992	55.58	1.2336	1.7031	97.09	0	9	18	48	304			
9	1.0608	1.5234	34.43	1.1079	1.668	58.31	1.211	1.6875	97.41	0	9	18	48	307			
10	0.9946	1.3984	37.5	1.0786	1.6445	56.35	1.1637	1.6484	98.1	0	9	18	48	311			
11	1.0796	1.5508	37.25	1.0594	1.6172	58.18	1.1207	1.6133	99.43	0	9	18	48	315			
12	1.0026	1.4805	39.17	1.0319	1.5937	58.25	1.0827	1.5742	100.39	0	9	18	48	319			
13	1.0127	1.5195	41.22	1.0096	1.5352	59.29	1.0679	1.543	101.18	0	9	18	48	322			
14	1.0271	1.4687	39.46	0.9898	1.5039	58.87	1.053	1.5273	101.56	0	9	18	48	325			
15	1.0146	1.4375	38.6	0.9729	1.4648	58.81	1.0408	1.457	98.63	0	9	18	48	329			
16	0.9959	1.4453	37.6	1.0495	1.6133	56.32	1.0281	1.4492	99.84	0	9	18	48	333			
17	0.9983	1.5117	36.67	0.9991	1.5117	59.19	1.0639	1.5	99.3	0	9	18	48	337			
18	0.9538	1.4883	38.16	0.985	1.4609	61.58	1.0957	1.5469	99.02	0	9	18	48	341			
19	0.9691	1.5469	39.29	0.9792	1.4687	64.23	1.1157	1.5586	97.04	0	9	18	48	344			
20	0.9833	1.5625	39.22	0.9796	1.4883	63.73	1.0921	1.5195	96.37	0	9	18	48	348			
21	0.9133	1.4961	43.71	1.0067	1.543	63.75	1.0709	1.5039	97.55	0	9	18	48	352			
22	0.9396	1.5508	44.35	1.0219	1.5898	62.94	1.0547	1.4766	97.66	0	9	18	48	356			
23	0.9109	1.4883	45.65	0.9548	1.4102	66.62	1.0532	1.457	97.67	0	9	18	48	359			
24	0.8947	1.4648	47.75	0.9023	1.3047	68.94	1.058	1.3906	95.08	0	9	18	48	363			
25	0.9018	1.4609	47.15	0.8944	1.332	66.31	1.0432	1.3437	94.29	0	9	18	48	367			
26	0.8734	1.4375	48.89	0.9032	1.3555	64.37	1.0221	1.3203	94.82	0	9	18	48	371			
27	0.874	1.4492	48.93	0.8494	1.3047	67.28	1.0038	1.3164	97.83	0	9	18	48	376			
28	0.8687	1.4687	49.98	0.8807	1.3828	65.36	1.0046	1.2891	99.4	0	9	18	48	379			
29	0.8504	1.4141	48.94	0.8348	1.2656	70.03	1.018	1.2578	99.5	0	9	18	48	383			
30	0.8072	1.3359	50.66	0.8829	1.375	66.81	1.0322	1.2461	99.81	0	9	18	48	386			
31	0.8603	1.4141	47.78	0.8612	1.3086	71.21	1.0563	1.2578	100.22	0	9	18	48	389			
32	0.8433	1.3945	45.62	0.8508	1.2969	70.7	1.0742	1.3125	101.75	0	9	18	48	393			
33	0.7803	1.2812	53.08	0.8498	1.2617	76.71	1.0716	1.3672	103.49	0	9	18	48	397			
34	0.8143	1.3359	54.86	0.8743	1.3164	76.3	1.0547	1.3984	105.25	0	9	18	48	401			
35	0.7524	1.1914	56.25	0.8468	1.2383	76.66	1.0372	1.4023	104.84	0	9	18	48	405			
36	0.8101	1.3086	51.18	0.8641	1.2773	80.11	1.042	1.4531	104.1	0	9	18	48	409			
37	0.8	1.2422	54.13	0.9036	1.3594	79.79	1.0525	1.5	104.4	0	9	18	48	413			

Fig. 4.6. Screenshot of the data acquisition

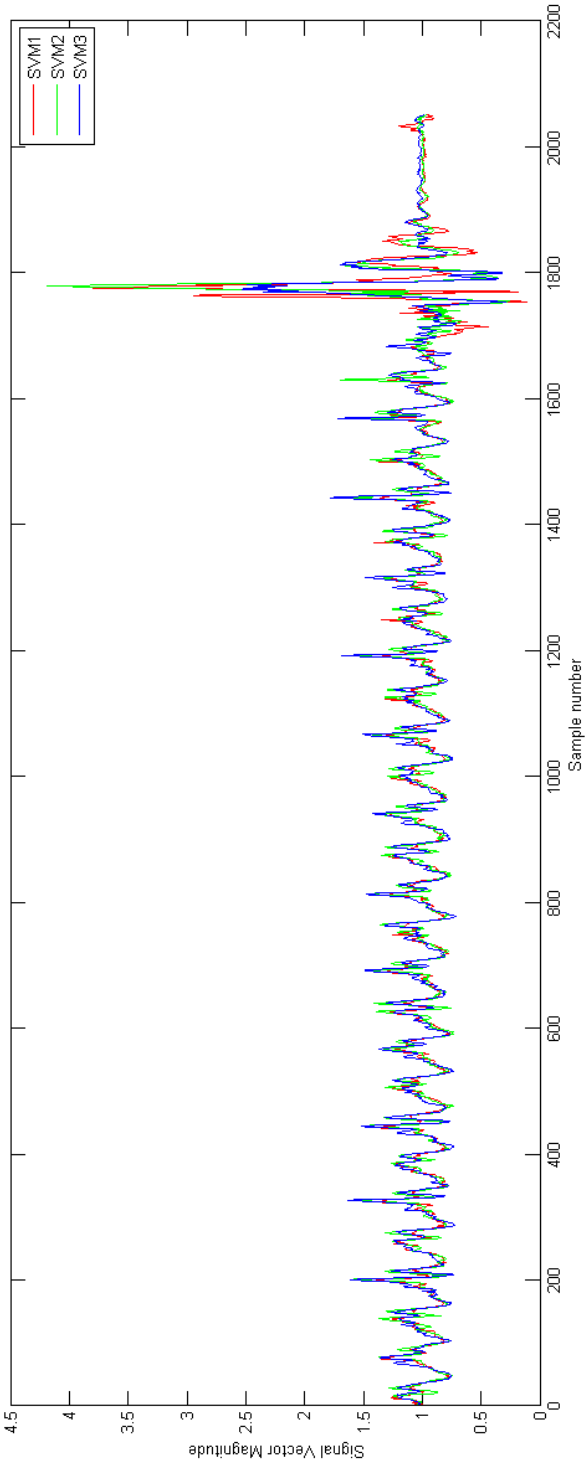


Fig. 4.7. Graph illustrating Front Fall

5. RESULTS

Three different tests were performed. The first two tests used two different resulting data sets from Test A and Test B described in the experimental protocol section and generated fall detection simulations using MATLAB. Prior to testing for fall detection accuracy, it was necessary to test the resulting decision tree thresholds on a series of consecutive samples of the fall data after the first time the threshold was met. Figure 5.2 shows accuracy results of fall detection using 5 to 25 consecutive samples after the first time the threshold is met. It was concluded that testing the threshold on 15 consecutive samples was the best option with about 86% accuracy. Once the test range was determined, the following tests were performed:

5.1 Test One

The data set generated using collected data in Test B was enrolled in the decision tree software. The output thresholds were then tested on data generated using Test B data. Figure 5.1 shows a fall detection simulation output. The green dot shows where the algorithm detected the fall. Fall detection classification was done as follows:

- A true positive occurs when a green dot lied before the lower most point of the plot as shown in Figure 5.1. We know that the fall trajectory occurs before the lowest peak based on the video images. In this case a fall was correctly identified. A fall positive occurs when a green dot appears in no fall data sets (i.e. tripping, sudden sitting). In this case no-fall data was incorrectly classified.

- A true negative occurs when a no-fall data set is correctly classified or when a green dot does not appear in no-fall data.
- A false negative occurs when a fall data set is incorrectly classified or when a green dot does not appear in a fall data set.

Fall detection accuracy results of Test One are represented in Figure 5.3.

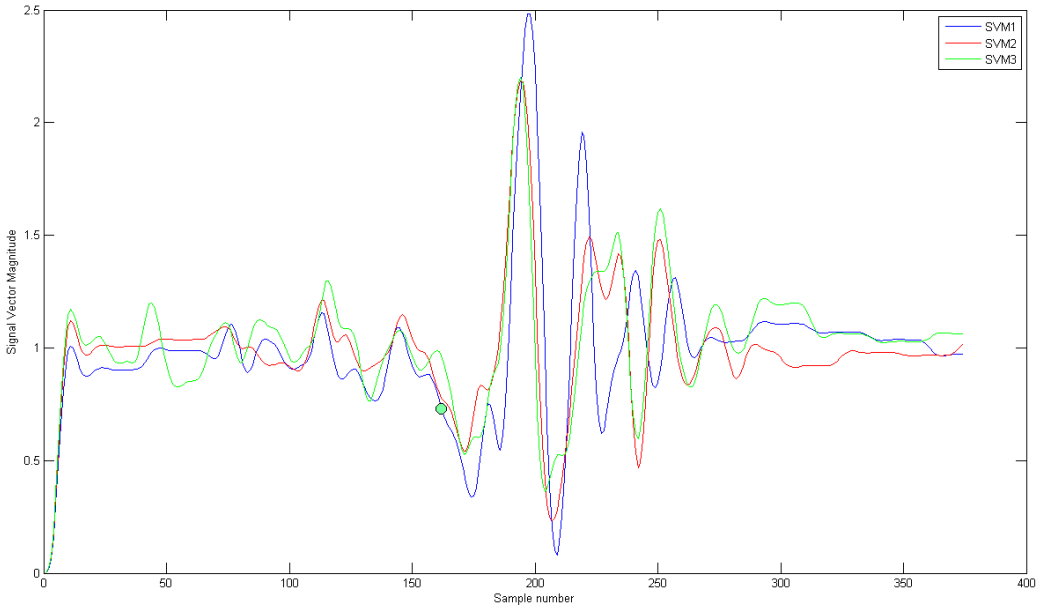


Fig. 5.1. Fall Detection Simulation (MATLAB Output Plot)

5.2 Test Two

The data set generated using collected data in Test B was enrolled in the decision tree software. The output thresholds were then tested on data generated using Test A data. Fall detection accuracy results for falls only of Test Two are represented in Figure 5.4. Figure 5.5 shows test results using thresholds on the data collected for a total of sixteen subjects.

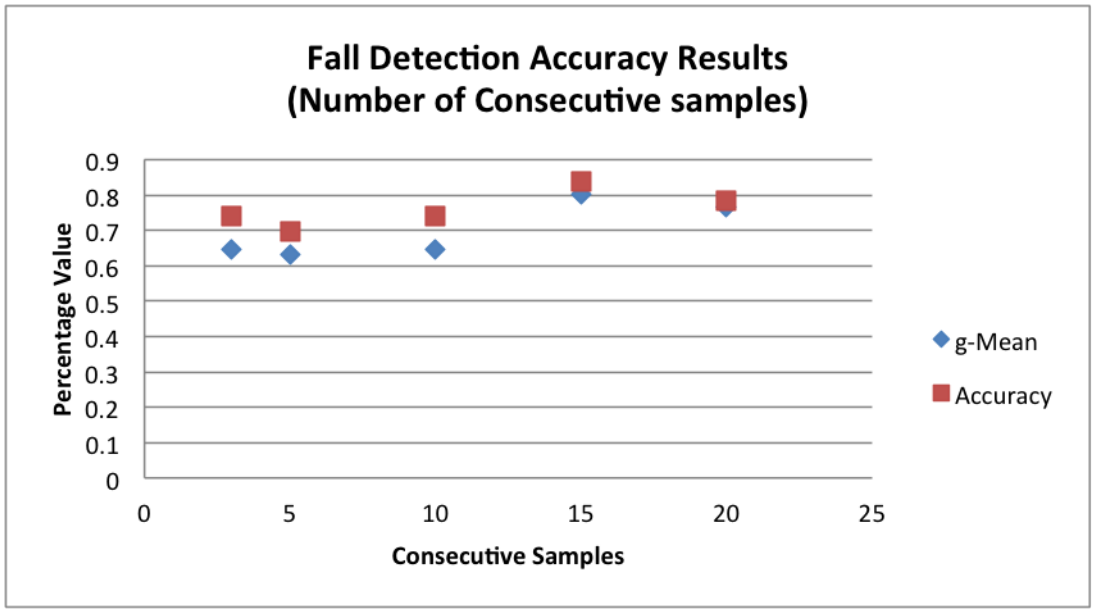


Fig. 5.2. Accuracy Results by Number of Consecutive Samples used for Testing

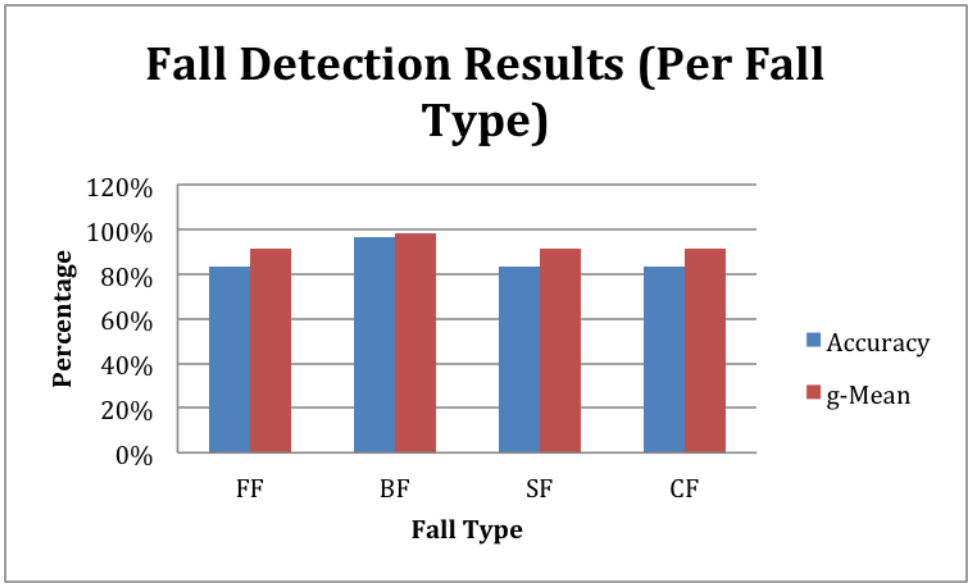


Fig. 5.3. Detection Results Enrolling and Testing with same Data Set (Data Set 2).

5.3 Test Three

In test three, the hardware prototype was tested. Fall detection was performed by the microprocessing unit in real time. Classification was recorded based on

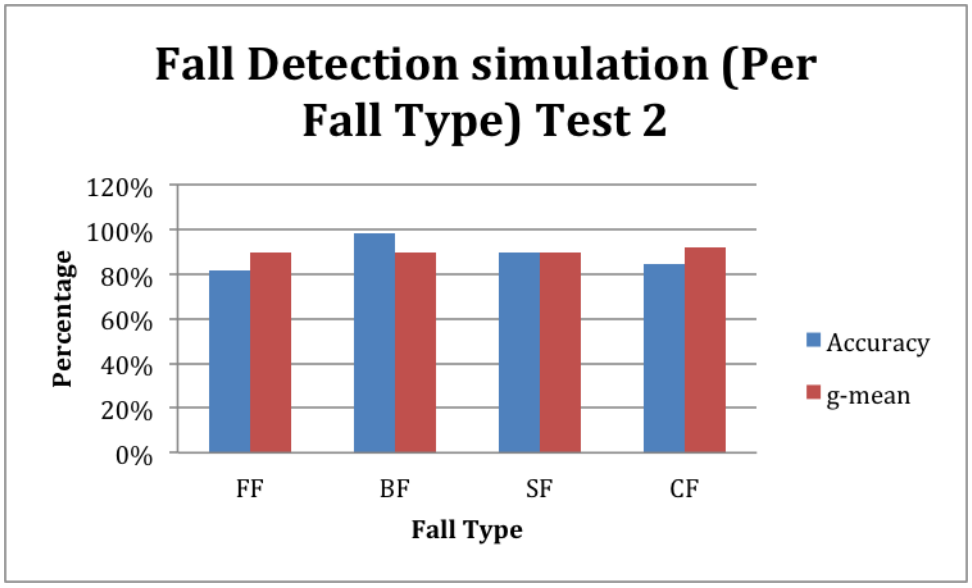


Fig. 5.4. Detection Results Enrolling Data set 2 and Testing on All Data (Data Set 1 and Data Set 2).

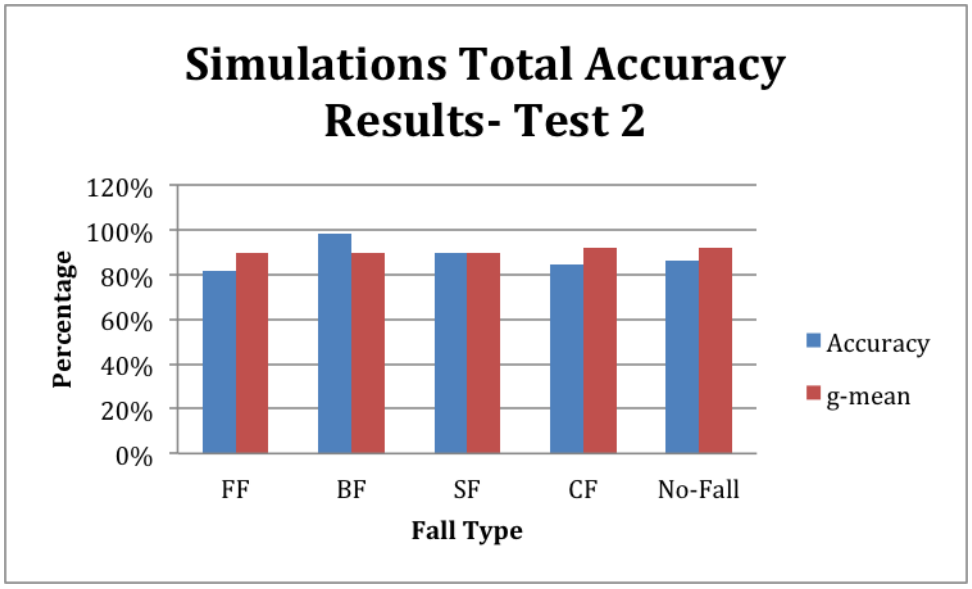


Fig. 5.5. Total Simulation Detection Results Enrolling Data set 2 and Testing on All Data

whether the LED light went on during falls or other no-fall events or movements. Using the output threshold values of the decision tree models and programming

them in the microprocessing unit, the prototype was tested in several frontal falls and data was collected. However, falls were not being detected under those threshold conditions. Using the new data of frontal falls generated by the hardware prototype, and observing the threshold values generated by the decision tree model, an informed selection of thresholds was performed as follows:

- For every fall, one SMA and one SVM, value for the three sensors were manually chosen from the range where fall happens (right before the lowest acceleration value). For simplicity and because high fluctuation of Tilt angle values, it was decided to only select SMA and SVM values.
- Out of all the falls, the lowest values of SMA, SVM, were selected. It was determined to select the lowest values because it would guarantee a closer threshold to the beginning of the fall.
- Manual thresholds were reprogrammed in the microprocessor.

Figures 5.6 and 5.7 show the results of test three.

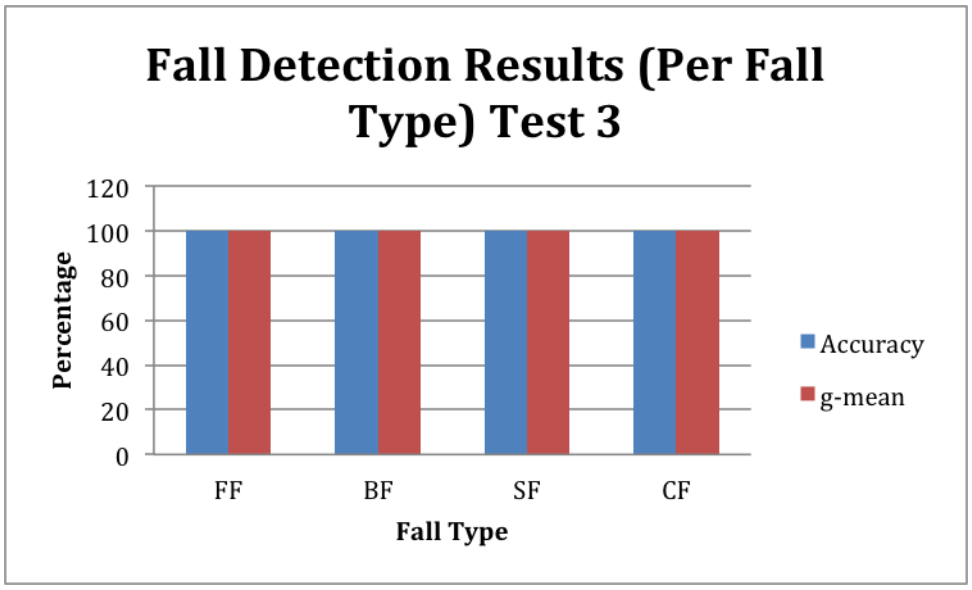


Fig. 5.6. Detection Results of Hardware Prototype for Falls Only

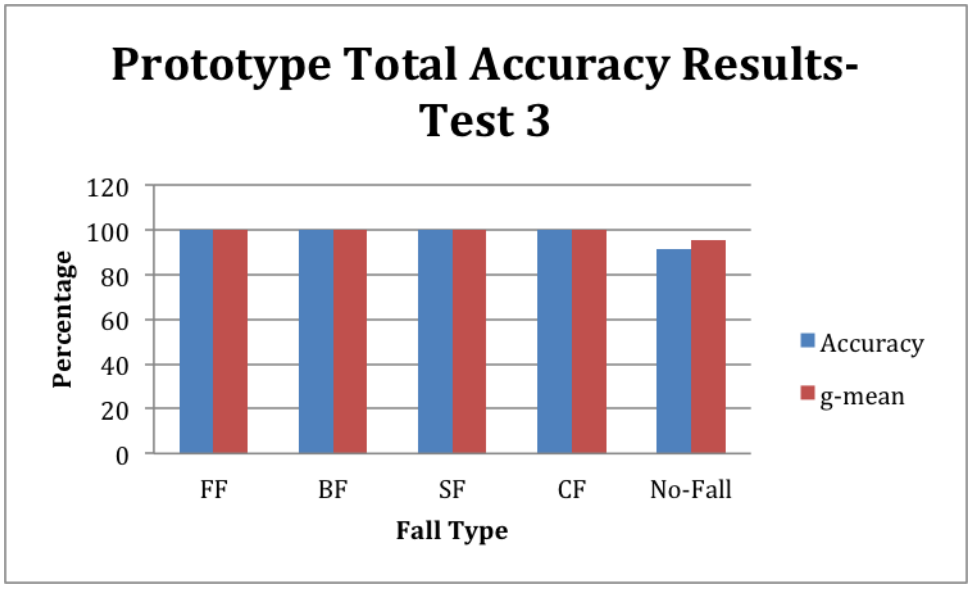


Fig. 5.7. Detection Results of Hardware Prototype including No-Falls

6. CONCLUSION AND FUTURE WORK

6.1 Conclusion

This thesis addresses how an integrated sensor system was designed for early fall detection in elders. In an initial phase, a deep understanding of neuroscience and the relationship between brain activity and fall events was developed through research. Then an off the shelf wireless sensor unit from Freescale (ZSTAR3) was used for data acquisition. For the initial set of experiments, a total of sixteen subjects performed seven different kinds of falls as well as no-fall activities. Data from the wireless triaxial accelerometers was used to calculate signal features like Signal Vector Magnitude, and Signal Magnitude Area, and Tilt Angle. These features were tested and simulated with MATLAB software, against each fall data set to determine thresholds, which were obtained using decision trees. Once the data was processed, a decision tree model was used to determine fall detection thresholds. A hardware prototype was then developed. This hardware features low power, high-speed sensors and processing units. The prototype, in which the calculated thresholds were programmed, was tested with a final set of experiments in which six volunteers were asked to imitate seven different kinds of falls while wearing the hardware prototype. Once again, the test included falls and non-falls. Accuracy was measured separately for total number of actual falls and total number of activities (which include both falls and non-falls). The new hardware prototype had an accuracy of 100% in detecting fall events and 95.55% accuracy in the case where all the fall and non-fall events are included. Figure 6.1 shows the closed loop functioning diagram of the project.

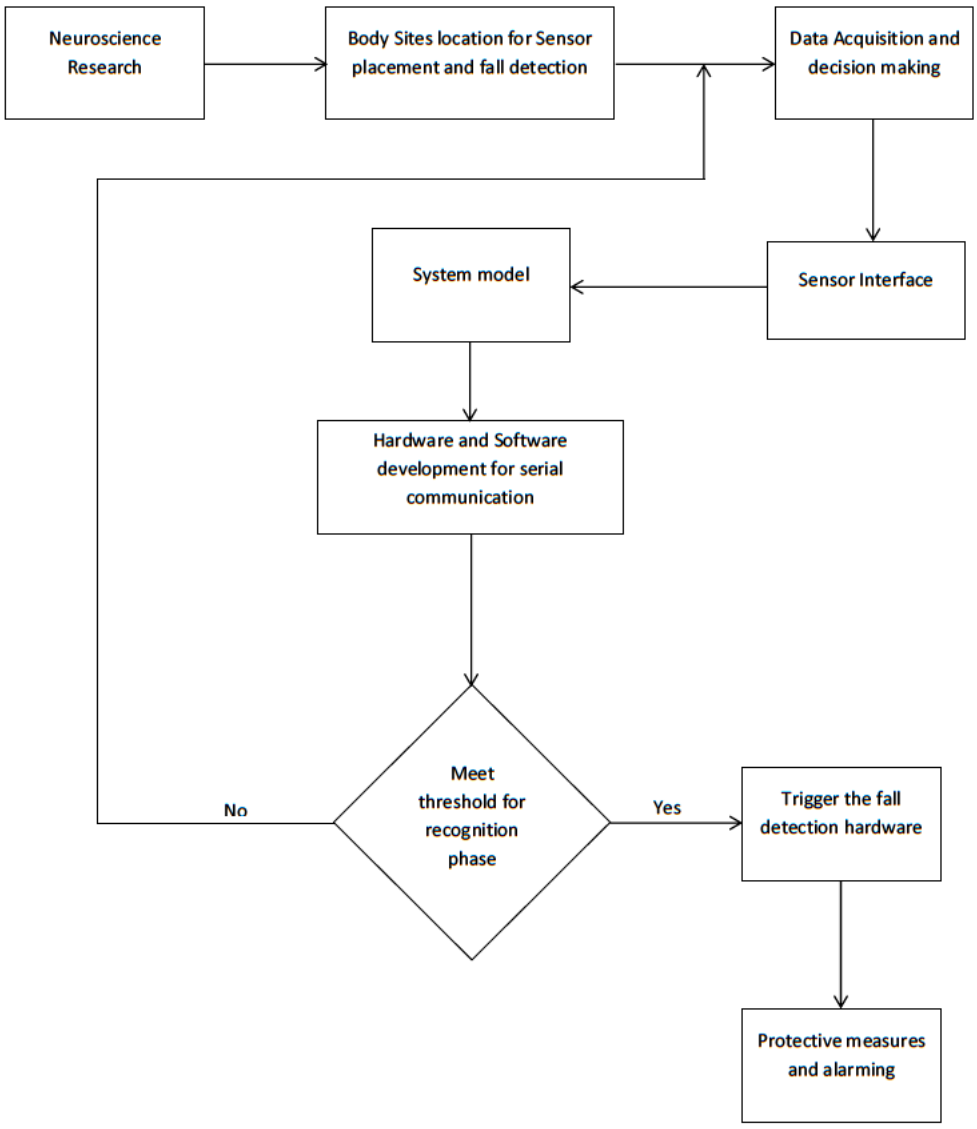


Fig. 6.1. Closed loop functioning diagram

6.2 Future Work

This thesis work shows an efficient working prototype of fall detection unit. However, it is important to eliminate further the occurrence of false positives. This can be refined by improving threshold values and by adding a gyroscope to classify both angular velocity and body position. Observing 84 samples per

second at the receiving end of the Bluetooth has brought sufficient resolution to detect the fall event, other sampling rates may be needed to try in order to optimize noise, calculation time, and robustness to achieve better real time application. This prototype can also be further extended by including a safety feature were airbags can be deployed using portable pressurized air cylinders to prevent hip and neck fractures during a fall event. Including a protection system requires further research in order to ensure the safety of the patient by elaborating not only a reliable system that will deploy timely, but also that it does not cause further injury to the patient. Research in determing the angle of impact will be helpful in deploying airbags in an intelligent way and it should be classified based on factors like height, weight and age. The system can also include a communication system that uses a cellphone application that integrates emergency services to assist people who have experienced a fall. The system can also include a log feature where data can be saved and used for further classification and specification of activities.

LIST OF REFERENCES

LIST OF REFERENCES

[1] M. E. Tinetti, W. L. Liu, and E. B. Claus, "Predictors and prognosis of inability to get up after falls among elderly people," *J. Am. Med. Assoc*, vol. 269, pp. 65–70, 1993.

[2] A. Shumway-Cook, MA.Ciol, J. Hoffman, B. Dudgeon, K. Yorston, and L. Chan, "Falls in the medicare population: incidence, associated factors, and impact on health care," *Physical Therapy*, vol. 89, no. 4, pp. 1–9, 2009.

[3] F. Englander, T. Hodson, and R. Terregrossa, "Economic dimensions of slip and fall injuries," *Journal of Forensic Science*, vol. 41, no. 5, pp. 733–46, 1996.

[4] J. Hausdorff, D. Rios, and H. Edelber, "Gait variability and fall risk in community-living older adults: a 1-year prospective study," *Archives of Physical Medicine and Rehabilitation*, vol. 82, no. 8, pp. 1050–6, 2001.

[5] M. Hornbrook, V. Stevens, D. Wingfield, J. Hollis, M. Greenlick, and M. Ory, "Preventing falls among community-dwelling older persons: results from a randomized trial," *The Gerontologist*, vol. 34, no. 1, pp. 16–23, 1994.

[6] B. Alexander, F. Rivara, and M. Wolf, "The cost and frequency of hospitalization for fall-related injuries in older adults," *American Journal of Public Health*, vol. 82, no. 7, pp. 1020–3, 1992.

[7] "Centers for disease control and prevention, national center for injury prevention and control," *Webbased Injury Statistics Query and Reporting System (WISQARS)*, November 2010.

[8] F. Bianchi, S. Redmond, M. Narayanan, S. Cerutti, and N. Lovell, "Barometric pressure and triaxial accelerometry-based falls event detection," *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, vol. 18, no. 6, pp. 619–627, Dec.

[9] T.-T. Nguyen, M.-C. Cho, and T.-S. Lee, "Automatic fall detection using wearable biomedical signal measurement terminal," in *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE*, pp. 5203–5206, Sept.

[10] F. Sposaro and G. Tyson, "ifall: An android application for fall monitoring and response," in *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE*, pp. 6119–6122, Sept.

[11] S. Sreelal, S. Sumadevi, P. Vinod, T. Varghese, S. Pillai, M. Pillai, and K. Damodaran, "A high resolution, integrated data acquisition system for aerospace applications," in *Digital Avionics Systems Conference, 2004. DASC 04. The 23rd*, vol. 2, pp. 13.E.3–131–10 Vol.2, 2004.

[12] Z. Escudero, M. Mai, and A. SantaFe, "Temperature sensor for medical applications based on erbium doped optical fiber," in *Engineering in Medicine and Biology Society, 2003. Proceedings of the 25th Annual International Conference of the IEEE*, vol. 4, pp. 3444–3445 Vol.4, 2003.

[13] M. Rahaman, M. Chowdhury, I. Nasir, and L.-T. Hwang, "Vsib: A sensor bus architecture for smart-sensor network," in *Computer Science and Information Engineering, 2009 WRI World Congress on*, vol. 3, pp. 436–439, 2009.

[14] Gray and Lincoln, "Vestibular system: Structure and function. neuroscience online. the university of texas health and science center," February 2013.

[15] Dougherty and Patrick, "Somatosensory systems. neuroscience online. the university of texas health and science center," February 2013.

[16] A. Bisdorff, A. Bronstein, C. Wolsley, M. Gresty, A. Davies, and A. Young, "Emg responses to free fall in elderly subjects and akinetic rigid patients.," *J Neurol Neurosurg Psychiatry*, vol. 66, no. 4, pp. 447–55, 1999.

[17] Freescale Semiconductor, *MMA8452Q 3-Axis, 12-bit/8-bit Digital Accelerometer*, 4.1 ed., 2011.

[18] Texas Instruments, *PCA9544A 4-Channel I2C and SMBus multiplexer with interrupt logic*, SCPS146D ed., 2008.

[19] D. Karantonis, M. Narayanan, M. Mathie, N. Lovell, and B. Celler, "Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 10, no. 1, pp. 156–167, Jan.

APPENDIX

APPENDIX: SOURCE CODE

```
#include <Wire.h>
#include <math.h>
#define MUX 0x70
#define SA0 1
#define MMA8452_ADDRESS 0x1D
const byte SCALE = 8;
const byte dataRate = 0;
int int1Pin = 2;
int int2Pin = 3;
int alarm = 12;
int accelCount[3];
float accelG[3];
float SVM1;
float SVM2;
float SVM3;
float SMA1;
float tilt1;
float SMA2;
float tilt2;
float SMA3;
float tilt3;
int t = 9999;
void setup()
{
```

```
Wire.begin();
Serial.begin(115200);
pinMode(int1Pin, INPUT);
digitalWrite(int1Pin, HIGH);
pinMode(int2Pin, INPUT);
digitalWrite(int2Pin, HIGH);
digitalWrite(alarm, LOW);
pinMode(alarm, OUTPUT);
byte c;
c = readRegister(0x0D);
if (c == 0x2A)
{
initMMA8452(SCALE, dataRate);
}
mux(0);
mux(1);
mux(2);
mux(3);
mux(0xFF);
}

void loop()
{
static byte source;
mux(0);
if (digitalRead(int1Pin)==1)
{
readAccelData(accelCount);
for (int i=0; i<3; i++)
```

```
accelG[i] = (float) accelCount[i]/((1<<12)/(2*SCALE));
SMA1 = abs(accelG[0]) + abs(accelG[1]) + abs(accelG[2]);
SVM1=sqrt( square(abs(accelG[0])) + square(abs(accelG[1])) + square(abs(accelG[2]))
);
tilt1 = acos(accelG[2] / SVM1) * 57.2957;
Serial.print(SVM1, 4);
Serial.print(",");
Serial.print(SMA1, 4);
Serial.print(",");
Serial.print(tilt1);
Serial.print(",");
}


mux(1);
if (digitalRead(int1Pin)==1)
{
readAccelData(accelCount);
for (int i=0; i<3; i++)
accelG[i] = (float) accelCount[i]/((1<<12)/(2*SCALE));
SMA2 = abs(accelG[0]) + abs(accelG[1]) + abs(accelG[2]);
SVM2=sqrt( square(abs(accelG[0])) + square(abs(accelG[1])) + square(abs(accelG[2]))
);
tilt2 = acos(accelG[2] / SVM2) * 57.2957;
Serial.print(SVM2, 4);
Serial.print(",");
Serial.print(SMA2, 4);
Serial.print(",");
Serial.print(tilt2);
Serial.print(",");
```

```
}

mux(2);
if (digitalRead(int1Pin)==1)
{
readAccelData(accelCount);
for (int i=0; i<3; i++)
accelG[i] = (float) accelCount[i]/((1<<12)/(2*SCALE));
SMA3 = abs(accelG[0]) + abs(accelG[1]) + abs(accelG[2]);
SVM3=sqrt( square(abs(accelG[0])) + square(abs(accelG[1])) + square(abs(accelG[2]))
);
tilt1 = acos(accelG[2] / SVM3) * 57.2957;
Serial.print(SVM3, 4);
Serial.print(",");
Serial.print(SMA3, 4);
Serial.print(",");
Serial.print(tilt3);
Serial.print(",");
}

if (SVM1<=0.744 && SMA1 <=0.9197 && SVM2<=0.8182 && SMA2<=0.9297
&& SVM3<=0.8543 && SMA3<=0.9414)
{
digitalWrite(alarm, HIGH);
Serial.print(t);
}
Serial.println();
mux(3);
mux(0xFF);
```

```
}

void mux(byte channel)
{
byte controlRegister = 0x04;
controlRegister |= channel;
Wire.beginTransmission(MUX);
if (channel == 0xFF)
{
Wire.write(0x00);
}
else
{
Wire.write(controlRegister);
}
Wire.endTransmission();
}

void readAccelData(int * destination)
{
byte rawData[6];
readRegisters(0x01, 6, rawData[0]);
for (int i=0; i<6; i+=2)
{
destination[i/2] = ((rawData[i] << 8)|rawData[i+1]) >>4;
if (rawData[i] > 0x7F)
{ destination[i/2] =  destination[i/2] + 1;
destination[i/2] *= -1;
}
```

```
}
}

void initMMA8452(byte fsr, byte dataRate)
{
MMA8452Standby();
if ((fsr==2)||(fsr==4)||(fsr==8))
writeRegister(0x0E, fsr >>2);
else
writeRegister(0x0E, 0);
writeRegister(0x2A, readRegister(0x2A)&(0x38));
if (dataRate <= 7)
writeRegister(0x2A, readRegister(0x2A) | (dataRate <<3));
writeRegister(0x11, 0x40);
writeRegister(0x13, 0x44);
writeRegister(0x14, 0x84);
writeRegister(0x12, 0x50);
writeRegister(0x21, 0x7F);
writeRegister(0x23, 0x20);
writeRegister(0x24, 0x20);
writeRegister(0x25, 0x08);
writeRegister(0x26, 0x30);
writeRegister(0x27, 0xA0);
writeRegister(0x28, 0xFF);
writeRegister(0x2C, 0x02);
writeRegister(0x2D, 0x19);
writeRegister(0x2E, 0x01);
MMA8452Active();
}
```

```
void MMA8452Standby()
{
byte c = readRegister(0x2A);
writeRegister(0x2A, c & (0x01));
}

void MMA8452Active()
{
byte c = readRegister(0x2A);
writeRegister(0x2A, c | 0x01);
}

void readRegisters(byte address, int i, byte * dest)
{
Wire.beginTransmission(MMA8452_ADDRESS);
Wire.write(address);
Wire.endTransmission(false);
Wire.requestFrom(MMA8452_ADDRESS, i);
int j = 0;
while(Wire.available())
{
dest[j] = Wire.read();
j++;
}
Wire.endTransmission();
}
```

```
byte readRegister(uint8_t address)
{
byte data;
Wire.beginTransmission(MMA8452_ADDRESS);
Wire.write(address);
Wire.endTransmission(false);
Wire.requestFrom(MMA8452_ADDRESS, 1);
data = Wire.read();
Wire.endTransmission();
return data;
}

void writeRegister(unsigned char address, unsigned char data)
{
Wire.beginTransmission(MMA8452_ADDRESS);
Wire.write(address);
Wire.write(data);
Wire.endTransmission();
}
```